

SAP HANA Cloud

Collaborative Database Development in SAP HANA Cloud, SAP HANA Database

Session 1: Project Lifecycle Management & HDI Concepts

SAP Data and Analytics
Virtual Forum:
Accelerating Outcomes

GET STARTED

WELCOME	4
ABOUT THIS SESSION	4
PREREQUISITES	5
INTRODUCTION	6
CHECK THE STATUS OF YOUR INSTANCE	7
1. CREATE AN SAP HANA DEVELOPMENT SPACE & PROJECT	9
CREATE A DEVELOPMENT SPACE	9
CREATE AN SAP HANA DATABASE PROJECT	11
ABOUT NAVIGATING THE SAP BUSINESS APPLICATION STUDIO ENVIRONMENT	13
2. CREATE A TABLE AND IMPORT DATA	14
CREATE AN HDBTABLE OBJECT	14
ADD THE COLUMN DEFINITIONS TO THE TABLE	16
ADD DATA TO THE INVENTORY TABLE	18
3. CHANGE DATA STRUCTURES USING HDBTABLE	22
4. CHANGE DATA STRUCTURES USING HDBMIGRATIONTABLE	24
RECREATE THE TABLE USING AN HDBMIGRATIONTABLE OBJECT	24
ADD ROWS TO THE HDBMIGRATIONTABLE OBJECT	26
5. CREATE HDBROLES	29
CREATE A ROLE WITH GRANT OPTION	29
CREATE A SECOND ROLE WITHOUT GRANT OPTION	32
DEPLOY THE ROLES TO YOUR DATABASE	32

SYNCHRONIZE YOUR PROJECT WITH GIT	33
CREATE A GITHUB TOKEN	34
CONNECT YOUR DEVELOPMENT PROJECT TO GITHUB	36
ADD A REMOTE ORIGIN VIA HTTPS CONNECTION	38
SUMMARY	39
ADDITIONAL RESOURCES	39

<https://community.sap.com/topics/hana-cloud>

© 2021 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.
The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions. Service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.

WELCOME

Join the SAP HANA Cloud Workshop series to learn from experts and follow along live using your own SAP HANA Cloud (trial) account.

Learn in this workshop how database development in SAP HANA Cloud and SAP Business Application Studio can be done collaboratively. You will get to know basic HDI concepts, ways to develop applications and to automate their deployment as well as creating template projects that others in your organization can easily use and modify.

About this session

Are you wondering what the best way is to manage changing data structures? Learn what benefits different table types in SAP HANA Cloud, SAP HANA database can offer you based on your needs. In this session, SAP Product Manager, Volker Saggau, will help you navigate through your challenges with the following tools and tips on how to:

- Set up a development space and a project in SAP Business Application Studio and connect it to your SAP HANA Cloud, SAP HANA database
- Use HDBTABLE objects to change data structures
- Use HDBMIGRATIONTABLE to change data structures
- Connect to Git

Your presenters:



Volker Saggau

Product Manager, SAP HANA Cloud | SAP



Jan Zwickel

Product Manager, SAP HANA Cloud | SAP

Prerequisites

Here is what you need to prepare before the workshop starts to be able to follow along:

1. [Sign up](#) for the free SAP HANA Cloud trial.
2. Provision an instance of SAP HANA Cloud, SAP HANA database and make sure it is running.
3. Subscribe to the free SAP Business Application Studio trial.
4. Subscribe to the CI/CD service in BTP cockpit.
5. Create a free account on www.github.com to use the required sample data.

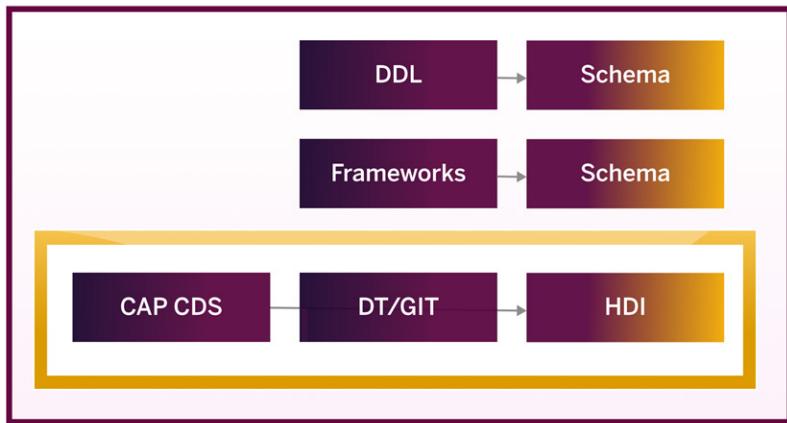
The steps needed to prepare are all covered in our preparation reader.

[Download it here](#)

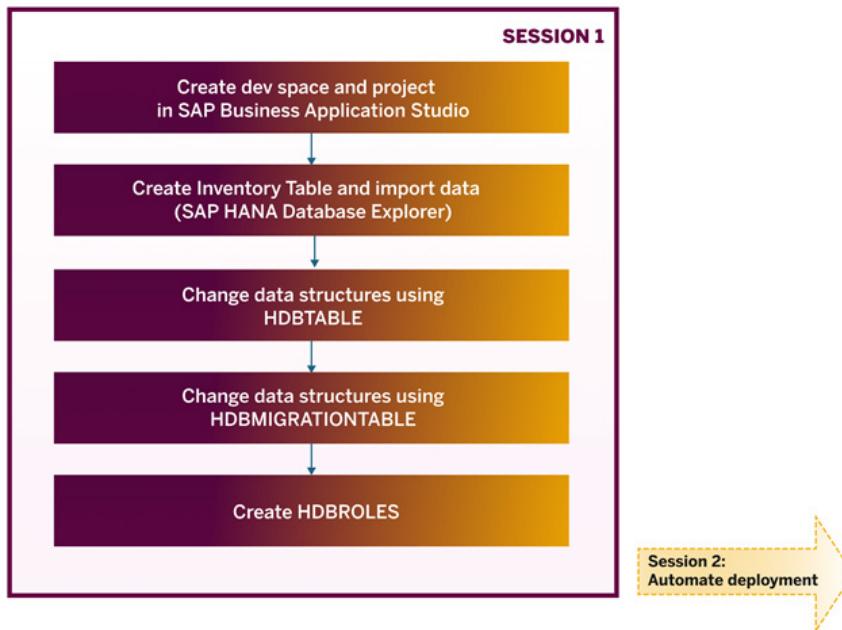
If you do, however, have any issues, please ask a question in the [SAP Community](#) using the tag SAP HANA Cloud or refer to the [SAP HANA Cloud Technical Documentation](#).

Introduction

In SAP HANA Cloud, Data Definition Language (DDL) can be used to create a table or a view in a schema. When you make changes to data structures in this case, you need to make sure that these changes are accepted by the database. Alternately, you can also use Frameworks (e.g. NetWeaver) to generate DDL and take responsibility for the changes made to data structures.



In this session, you will generate design-time artifacts (or objects) to manage the data present in an HDI container. This is done using the support of a GitHub repository. You can manage changing data structures using two design-time object types: HDBTABLE and HDBMIGRATIONTABLE. This session will include both the approaches while also highlighting the advantages of using an HDBMIGRATIONTABLE over HDBTABLE.



In the next session, we will use CAP CDS to generate the design-time objects and deploy them to the database. This is done to automate the process of changing data structures in a database. Cloud Application Programming Core Data Services (CAP CDS) is an abstract language used to describe how your data structures should look like.

CHECK THE STATUS OF YOUR INSTANCE

Before you can start with the steps of this session, you need to make sure that your SAP HANA Cloud, SAP HANA database instance is running.

1. Open the trial homepage in SAP BTP cockpit.
2. Click on **Go To Your Trial Account** from the SAP BTP Trial homepage.

The screenshot shows the SAP BTP Cockpit interface. At the top, it says "Welcome to SAP BTP Trial". Below that, there's a button labeled "Go To Your Trial Account". Underneath, there's a section titled "Quick Tool Access" with three items: "SAP Business Application Studio" (with a icon of a blue square), "CLI for BTP" (with a icon of two arrows), and "APIs for SAP BTP" (with a icon of a gear).

3. In SAP BTP Cockpit, select your **Subaccount** and within the subaccount, select your **Space**.

The screenshot shows the SAP BTP Cockpit interface with the sidebar open. Under "Subaccount: trial", it shows "Active Subscriptions" with 4 available. The "Cloud Foundry Environment" section is expanded, showing details like API Endpoint, Org Name, and Org ID. The "Spaces (1)" table has one row named "dev" with 1 application and 6 service instances. The "Kyma Environment" section is also visible.

Name	Applications	Service Instances
dev	1	6

4. After selecting your Space, click on the **SAP HANA Cloud** option on the panel on the left-side of the screen.

The screenshot shows the SAP BTP Cockpit interface. On the left sidebar, under the 'Instances' section, the 'SAP HANA Cloud' option is highlighted with a yellow box. The main content area displays a table for the 'Space: dev - Applications'. The table has columns: Requested State, Name, Instances, Disk Quota, Memory, and Actions. There is one entry: 'BindMobileApplicationRoutesToME' with a 'Stopped' state, 0/1 instances, 256 MB disk quota, 128 MB memory, and a row of icons for Actions.

5. On the tile of your trial SAP HANA database instance click on **Actions**.
 6. Select **Open in SAP HANA Cloud Central**. To proceed to SAP HANA Cloud Central, you might need to sign in with your user credentials now.

The screenshot shows the SAP BTP Cockpit interface. The 'SAP HANA Cloud' option is selected in the sidebar. A specific instance named 'DEMO' is selected, showing its details: Memory (30 GB), CPU (2 vCPUs), and Storage (120 GB). A context menu is open over the instance, with the 'Actions' option highlighted with a yellow box. The menu items include: Copy SQL Endpoint, Copy Instance ID, Open in SAP HANA Cloud Central (which is also highlighted with a yellow box), Open in SAP HANA Cockpit, Open in SAP HANA Database Explorer, and Open SQL Console.

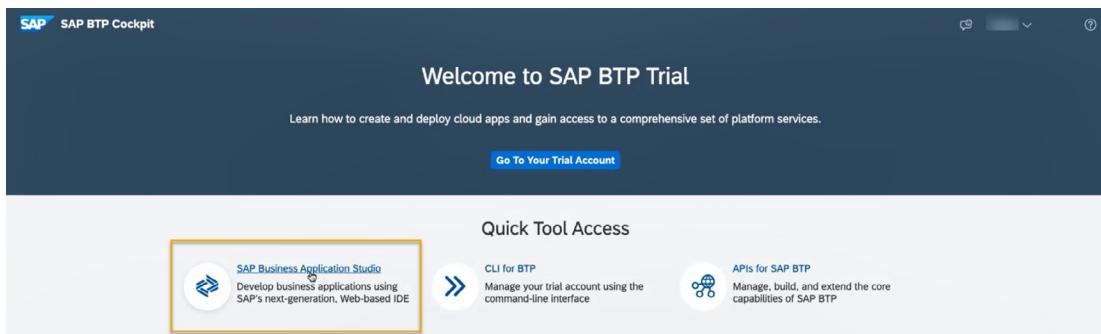
7. In SAP HANA Cloud Central, check the status of your database instance from the list of all instances. Ensure it has the status **RUNNING**, otherwise click on the **three dots** icon in the column **Actions** and select Start.

The screenshot shows the SAP HANA Cloud Central interface. The left sidebar shows 'Instances'. The main area displays a table of instances. One instance named 'DEMO' is listed with the following details: Status (RUNNING, highlighted with a yellow box), Name (DEMO), Type (SAP HANA Database), Notifications, Storage (120 GB Storage (1 node)), Compute (2 vCPUs (1 node)), Memory (30 GB Memory (1 node)), and Actions (a three-dot menu icon highlighted with a yellow box). Filter options at the top include Organization, Space, Status, Alert, and Type (Data Lake x 1 More).

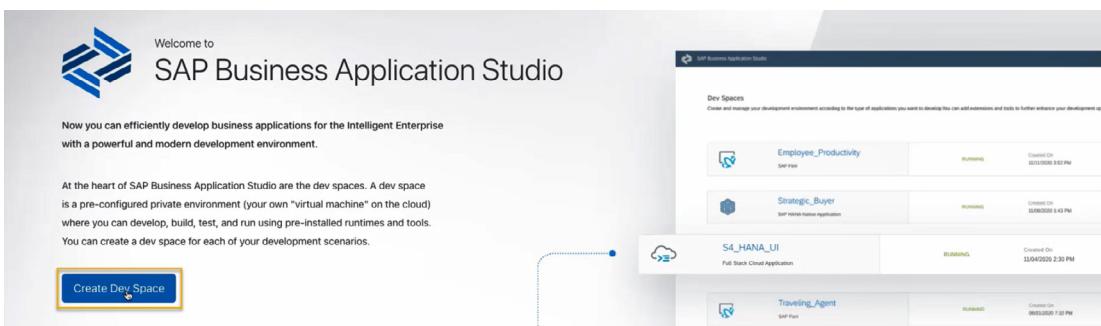
1. CREATE AN SAP HANA DEVELOPMENT SPACE & PROJECT

Create a development space

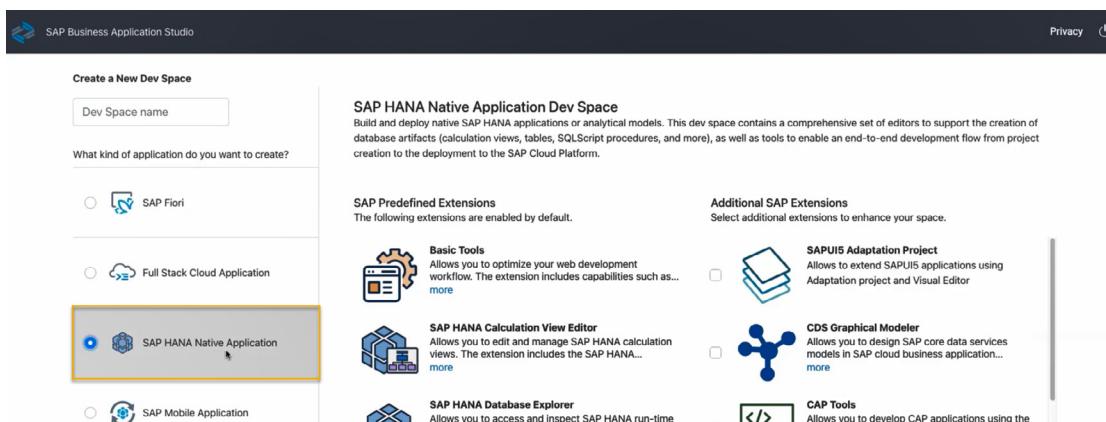
1. Open **SAP Business Application Studio**. In a trial landscape you can get there via the **Trial Home** page in SAP BTP cockpit.



2. A new window opens for SAP Business Application Studio. Click on **Create Dev Space**.



3. In the **Create a New Dev Space** window, select **SAP HANA Native Application Dev Space** as the option for the kind of application you want to create. This will add SAP Predefined Extensions to your development space that allow you to deploy native SAP HANA applications or analytical models.



4. Next, name your dev space in the field in the top left corner, for example as **WS3_1**. This name must start with a letter or number and only allows alphanumeric characters and underscores, no special characters. You can see if the name is valid if a green tick mark becomes visible alongside the name field.

5. Here, you could also select additional SAP Extensions for your dev space. For the purpose of completing this workshop session, you do **not** need to select any of the extensions.
6. Finally, click on **Create Dev Space** at the bottom right corner of the screen.
7. A new window opens for Dev Spaces. Here, you can view all the existing Dev Spaces you have created.

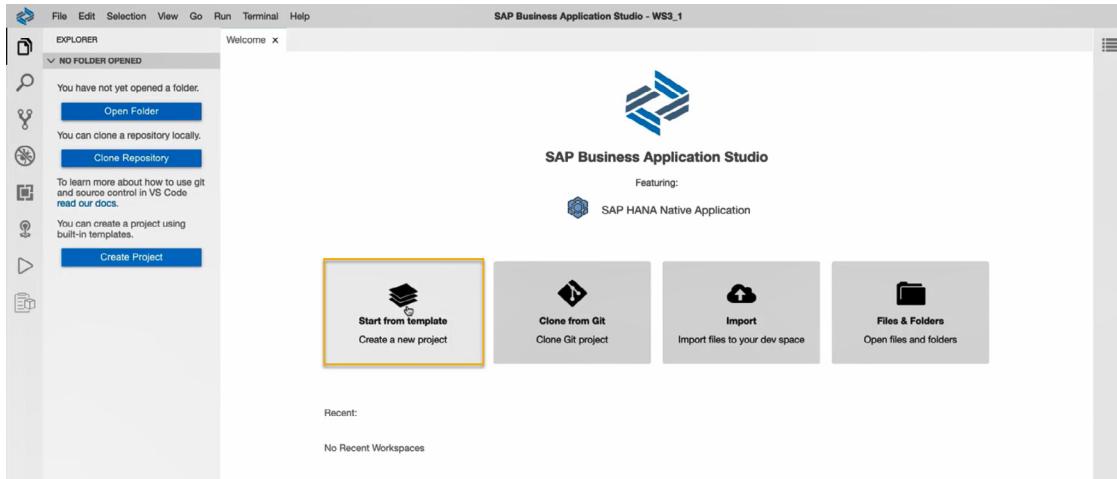
Important: As a trial user, you can have a maximum of two Dev Spaces while only having one of them running. Throughout this workshop series, you will create a second dev space in session 2. In session 3, you can re-use the dev space you just created. If you have other dev spaces before starting this workshop, we recommend you remove them. Alternatively, if you have an SAP HANA Native Application Dev Space, you can create a new project inside that space.

8. The process of creation of development space usually takes a few minutes.
9. Once the status changes to **RUNNING**, you can click on the name of your dev space to open it.

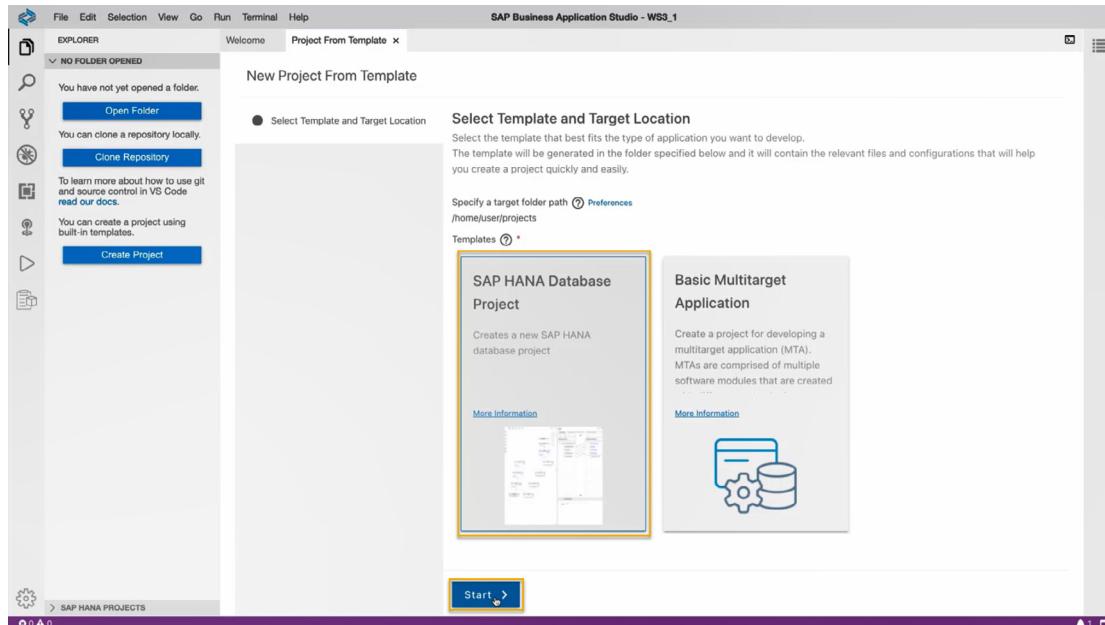
WS3_1	RUNNING	Created On 07/27/2021 10:57 AM	ID ws-bjqr	Disk Usage Currently unavailable	Action Icons
SAP HANA Native Application					

Create an SAP HANA Database Project

- After you have opened your dev space in SAP Business Application Studio, select **Start from template** from the **Welcome** screen to create a new project.

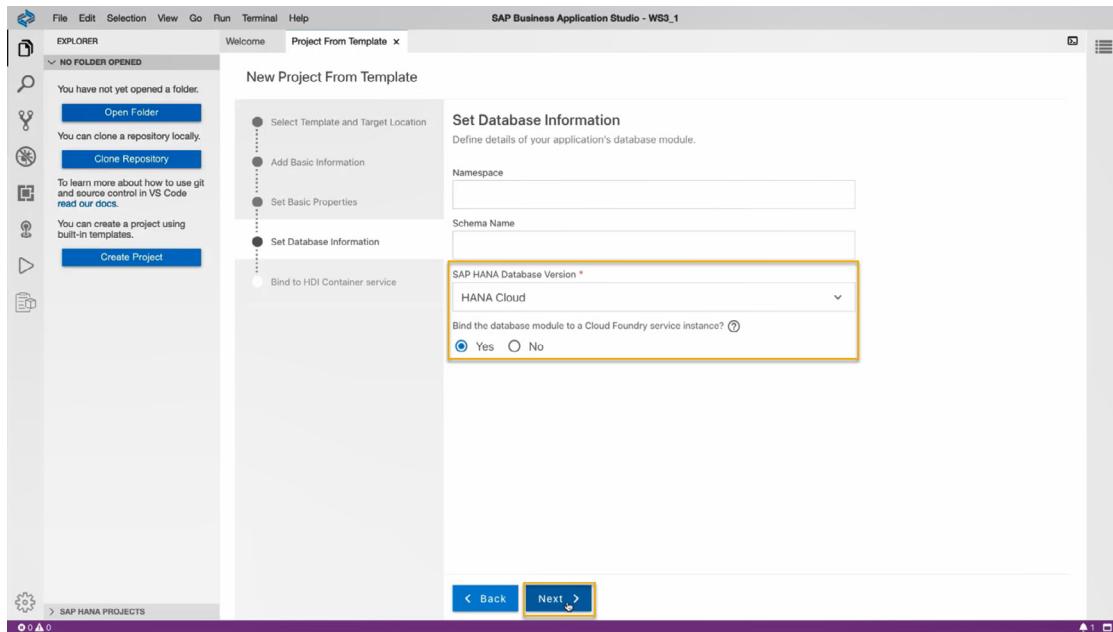


- Choose the **SAP HANA Database Project** template and click on **Start**.

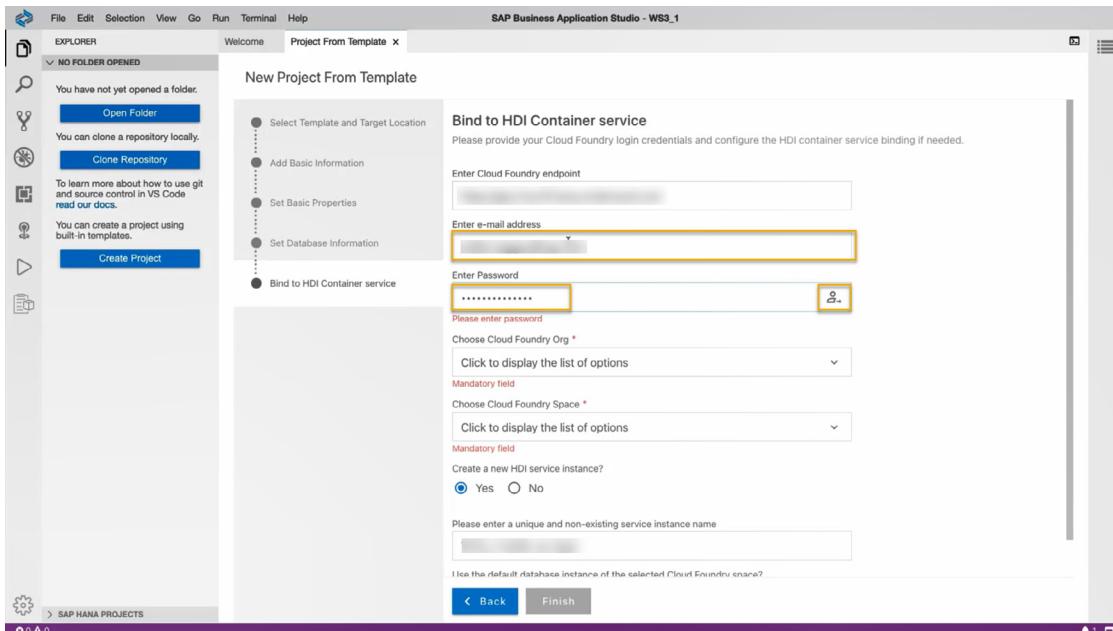


- In the step **Add Basic Information**, add the name of your new Project as **WS3_1**. Click on **Next**.
- In the step **Set Basic Properties**, keep the database module name as **db**. Click on **Next**.
- In the step **Set Database Information**, we recommend **not** to fill in the fields **Namespace** and **Schema** Name for the purpose of this workshop.

6. Set SAP HANA Database Version as HANA Cloud.



7. Then, select **Yes** for the option '**Bind the database module to a Cloud Foundry service instance?**'
8. Click on **Next**.
9. In the step **Bind to HDI Container service**, enter your email address (the one you used for your SAP HANA Cloud trial account) and corresponding password to make the project target your trial account.
10. Then click on the login icon. This will automatically select your Cloud Foundry Organization and Cloud Foundry Space.



11. Select **Yes** for both the options '**Create a new HDI service instance?**', and '**Use the default database instance of the selected Cloud Foundry space?**'. A unique and non-existing service instance name will be generated automatically for you.
12. Finally, click on **Finish** and your project will be created. This process usually takes a moment to complete.

About navigating the SAP Business Application Studio Environment

On the left panel, you can now see your workspace where you have your files, and the SAP HANA Project panel where you can deploy your project and open the HDI container in the SAP HANA Database Explorer. Whenever you add a new database object that can be deployed to your HDI container, it will appear in this panel and you can deploy it. In this area you can also check the database connections of your project.

Moving your cursor to the name of an object, folder, or connection in this panel, you will see different icons:

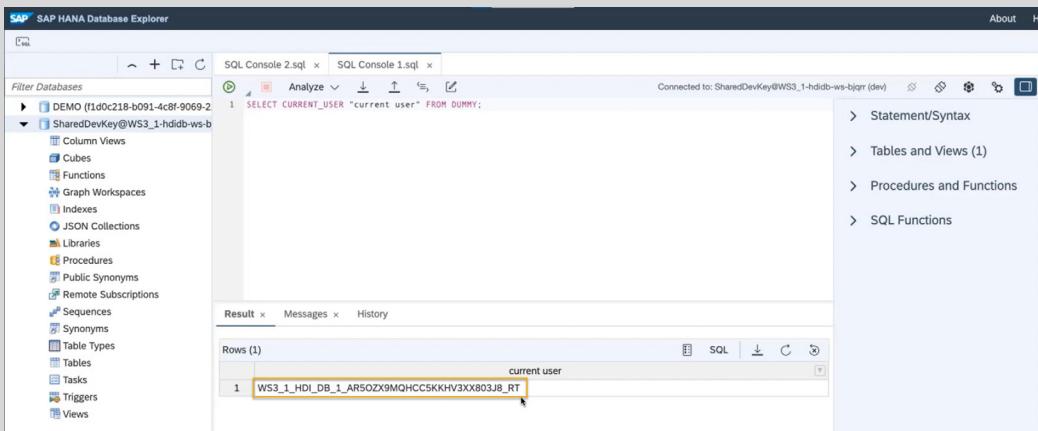
- This icon  will open the HDI container in SAP HANA Database Explorer
- This icon  will deploy an object, folder or the whole project to the HDI container or other connected database.
- This icon  (only on the Database Connections level) allows you to create a new database connection.
- This icon  allows you to bind (green colour) or unbind (grey colour) a database connection.

Note: When you are opening the HDI container in the SAP HANA Database Explorer, you will automatically be connected as the run-time user and this connection will be added to your list of database connections. This is different to opening the SAP HANA Database Explorer via the SAP BTP cockpit or SAP HANA Cloud Central, which would establish a connection using the DBADMIN user.

To check what user you are currently connected with, you can use the following statement in a SQL console:

```
SELECT CURRENT_USER "current user" FROM DUMMY;
```

Using this statement in the HDI container connection opened via SAP Business Application Studio will show the run-time (RT) user of this HDI container:

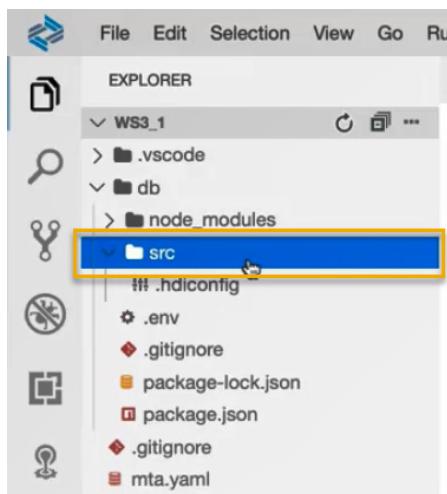


2. CREATE A TABLE AND IMPORT DATA

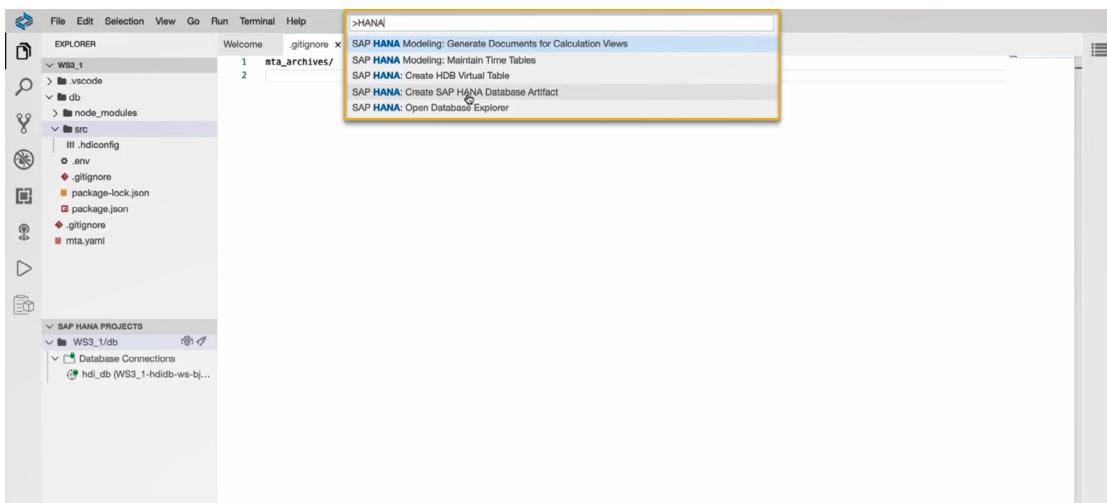
Create an HDBTABLE Object

Now you can create the first object in your project: an HDBTABLE that will store the sample data used in this workshop.

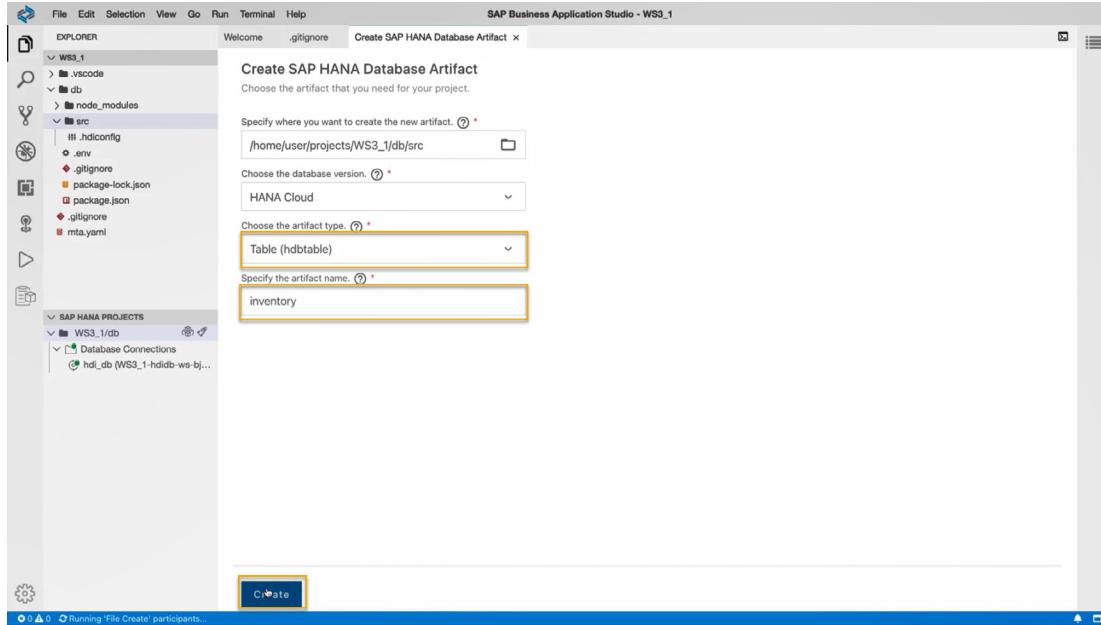
- In the **EXPLORER** panel on the left side, expand the folders under your Project **WS3_1**. Click on the **src** folder.



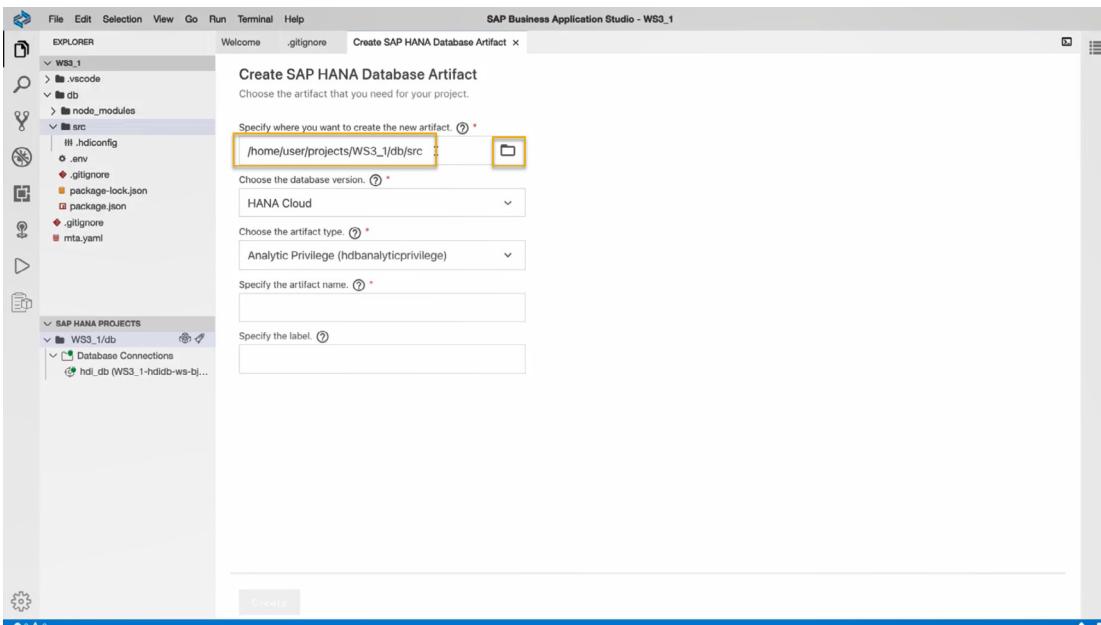
- To create a new SAP HANA Database Artifact, you can press the **F1 key** or click on **View – Find Command** at the top of the screen to open a search prompt. Enter **HANA** in the field and from the dropdown list of functionalities, choose **SAP HANA: Create SAP HANA Database Artifact**.



3. In the window for creating a new SAP HANA Database Artifact, click on the folder icon in the first field (Choose path where you want to create the artifact). Navigate to your project, the **src** folder, and then click on **Open**.



4. Select the artifact type as **Table (hdbtable)**.
 5. In the field ‘Specify the artifact name’ enter ‘**inventory**’.
 6. The database version should be HANA Cloud, by default. Without making any further changes, click on **Create**.



7. A new file **inventory.hdbtable** can be seen under the **src** folder in the EXPLORER panel.

Add the column definitions to the table

- Click on the **inventory.hdbtable** file to open the file window. You can see that the file only contains the code: column table "inventory" (). Remove the default code in this file and paste the code below in this file to add the column definitions to this new table.

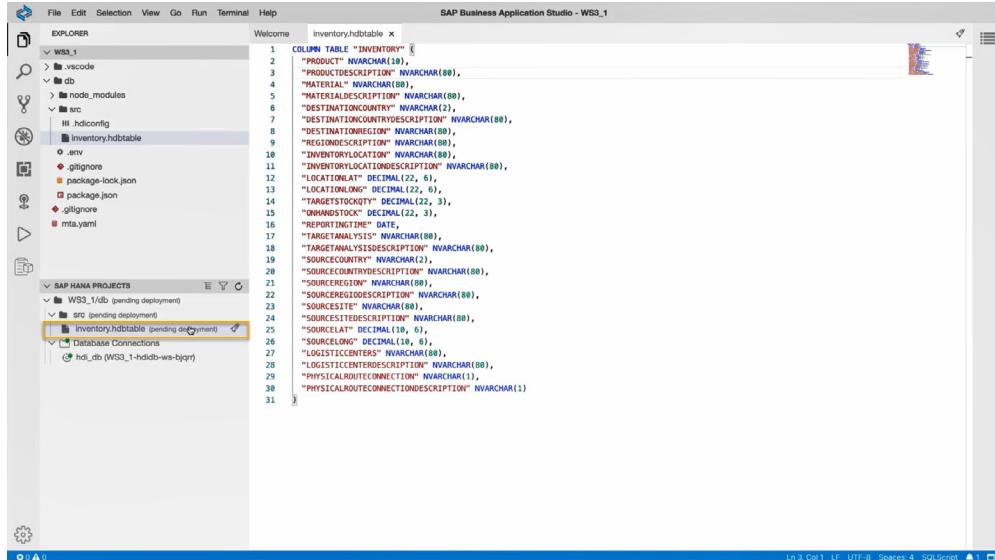
Important: The name of the table in the code has to be written in **upper case characters**.

```
column table "INVENTORY" (
    "PRODUCT" NVARCHAR(10),
    "PRODUCTDESCRIPTION" NVARCHAR(80),
    "MATERIAL" NVARCHAR(80),
    "MATERIALDESCRIPTION" NVARCHAR(80),
    "DESTINATIONCOUNTRY" NVARCHAR(2),
    "DESTINATIONCOUNTRYDESCRIPTION" NVARCHAR(80),
    "DESTINATIONREGION" NVARCHAR(80),
    "REGIONDESCRIPTION" NVARCHAR(80),
    "INVENTORYLOCATION" NVARCHAR(80),
    "INVENTORYLOCATIONDESCRIPTION" NVARCHAR(80),
    "LOCATIONLAT" DECIMAL(22, 6),
    "LOCATIONLONG" DECIMAL(22, 6),
    "TARGETSTOCKQTY" DECIMAL(22, 3),
    "ONHANDSTOCK" DECIMAL(22, 3),
    "REPORTINGTIME" DATE,
    "TARGETANALYSIS" NVARCHAR(80),
    "TARGETANALYSISDESCRIPTION" NVARCHAR(80),
    "SOURCECOUNTRY" NVARCHAR(2),
    "SOURCECOUNTRYDESCRIPTION" NVARCHAR(80),
    "SOURCEREGION" NVARCHAR(80),
    "SOURCEREGIODESCRIPTION" NVARCHAR(80),
    "SOURCESITE" NVARCHAR(80),
    "SOURCESITEDESCRIPTION" NVARCHAR(80),
    "SOURCELAT" DECIMAL(10, 6),
    "SOURCELONG" DECIMAL(10, 6),
    "LOGISTICCENTERS" NVARCHAR(80),
    "LOGISTICCENTERDESCRIPTION" NVARCHAR(80),
    "PHYSICALROUTECONNECTION" NVARCHAR(1),
    "PHYSICALROUTECONNECTIONDESCRIPTION" NVARCHAR(1)
)
```

Note: You can right-click inside the file window and choose **Format document** to clean up the formatting of the file.

2. On the bottom left corner, you can see in the **SAP HANA Projects** panel that the table has been added.

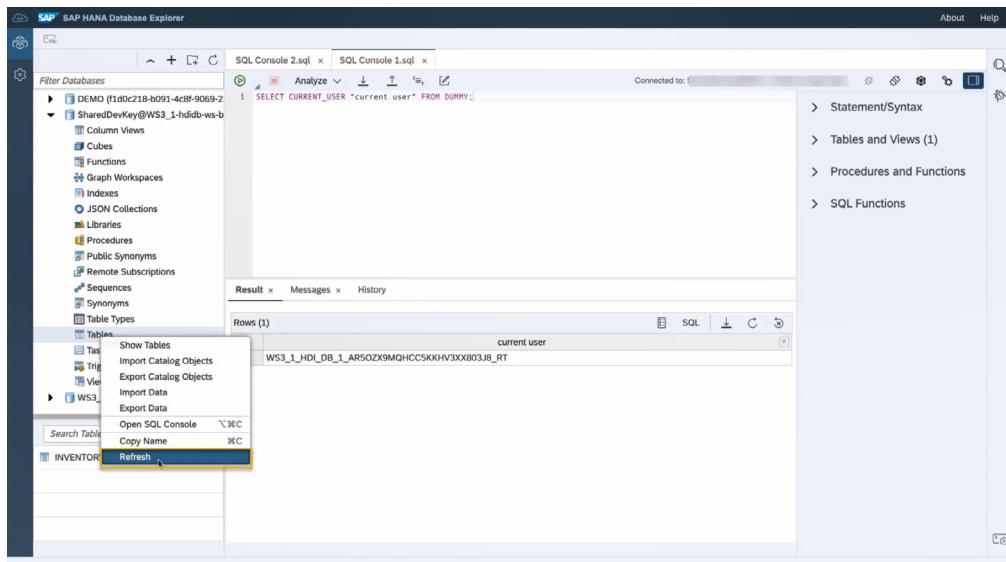
Click on the  deploy icon next to the **inventory.hdbtable** to deploy the table to your **hdi container**. You can check in the terminal if the table was successfully deployed. Alternately, you can also deploy by clicking on the deploy icon on the project level (WS3_1/db).



3. In the SAP HANA Project panel, click on open the  HDI container icon. The **SAP HANA Database Explorer** will open in a new tab.



4. In the catalog of the **HDI container connection**, click on **Tables**. If you cannot see the INVENTORY table, right-click on **Tables** and select **Refresh**.



- Click on the **INVENTORY** table to view the list of all the columns you have just defined.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there's a tree view of database objects under 'Filter Databases'. A yellow box highlights the 'Tables' node. Below it, a search bar shows 'INVENTORY'. Another yellow box highlights the 'INVENTORY' table in the results list. The main pane displays the table structure with 29 columns. The columns are listed in rows, with each row containing the column name, SQL data type, key status, not null status, default value, and comment. The first few columns are: TARGETSTOCKQTY (DECIMAL(22,3)), ONHANDSTOCK (DECIMAL(22,3)), REPORTINGTIME (DATE), TARGETANALYSIS (NVARCHAR(80)), and TARGETANALYSISDESCRIPTION (NVARCHAR(80)).

Name	SQL Data Type	Key	Not Null	Default	Comment
13 TARGETSTOCKQTY	DECIMAL(22,3)			NULL	
14 ONHANDSTOCK	DECIMAL(22,3)			NULL	
15 REPORTINGTIME	DATE			NULL	
16 TARGETANALYSIS	NVARCHAR(80)			NULL	
17 TARGETANALYSISDESCRIPTION	NVARCHAR(80)			NULL	
18 SOURCECOUNTRY	NVARCHAR(2)			NULL	
19 SOURCECOUNTRYDESCRIPTION	NVARCHAR(80)			NULL	
20 SOURCEREGION	NVARCHAR(80)			NULL	
21 SOURCEREGIONDESCRIPTION	NVARCHAR(80)			NULL	
22 SOURCESITE	NVARCHAR(80)			NULL	
23 SOURCESITEDESCRIPTION	NVARCHAR(80)			NULL	
24 SOURCELAT	DECIMAL(10,6)			NULL	
25 SOURCELONG	DECIMAL(10,6)			NULL	
26 LOGISTICCENTERS	NVARCHAR(80)			NULL	
27 LOGISTICCENTERDESCRIPTION	NVARCHAR(80)			NULL	
28 PHYSICALROUTECONNECTION	NVARCHAR(1)			NULL	
29 PHYSICALROUTECONNECTIONDESCR	NVARCHAR(1)			NULL	

If you click on **Open Data**, you can see that the table does not contain any data yet. Let's change that!

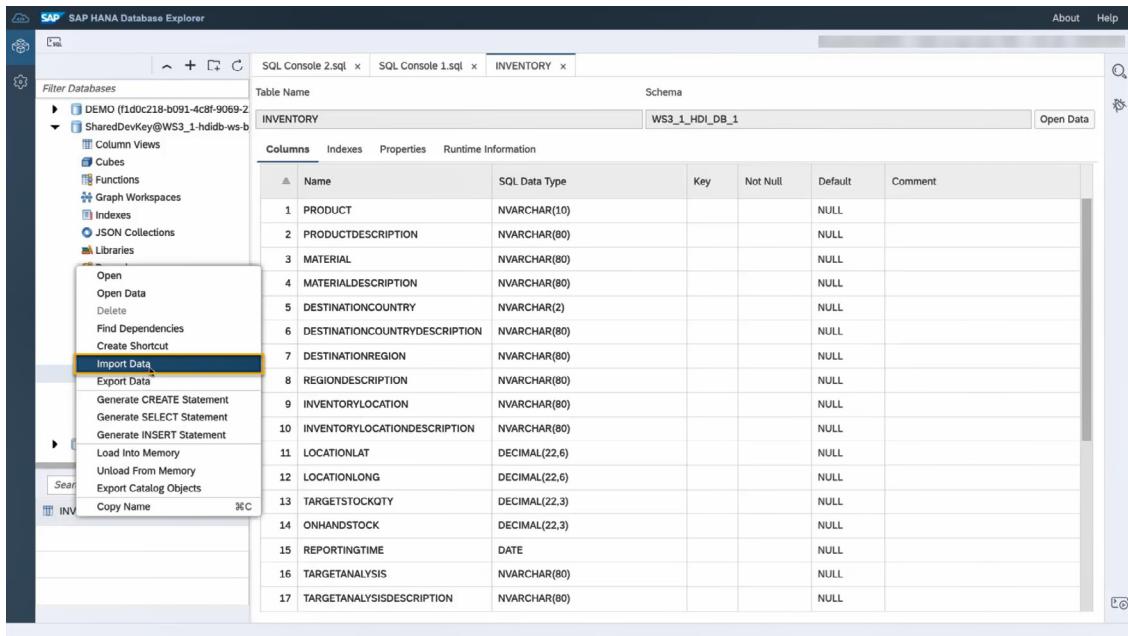
Add data to the Inventory table

- To download the sample data from the GitHub repository, click [here](#). Select the **Download** option for the **Sample Data.zip** file under the folder location SAP-samples/hana-cloud-learning/Workshop: Collaborative Development.

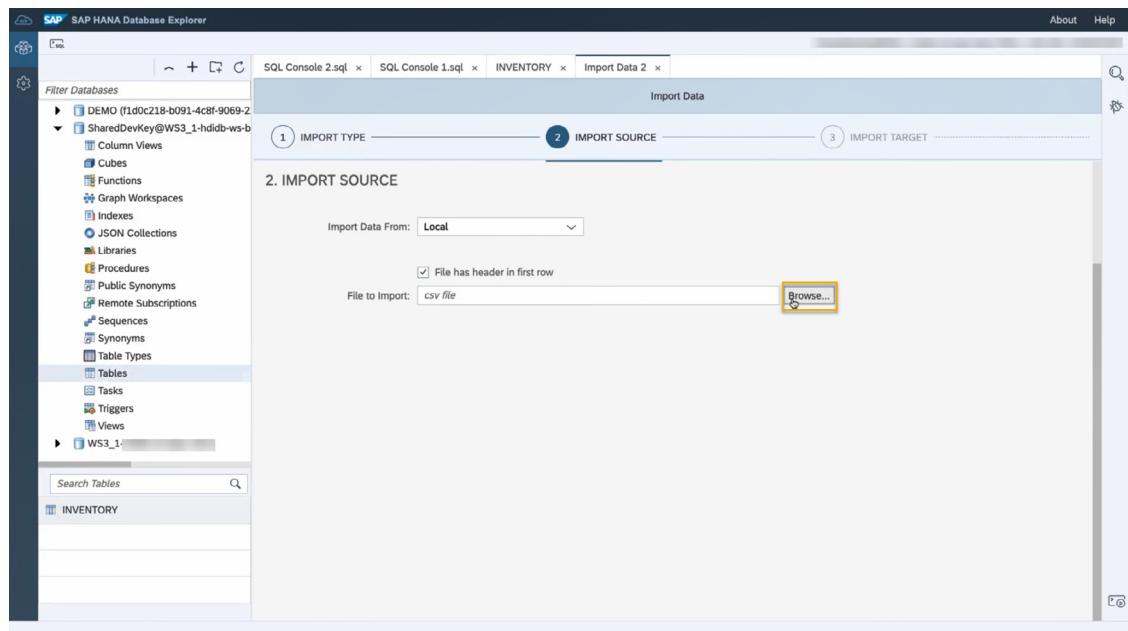
The screenshot shows a GitHub repository page for 'SAP-samples / hana-cloud-learning'. The repository was generated from 'SAP-samples/apache2-reuse-template'. It has 5 stars, 0 forks, and 1 contributor named 'chriskollhed'. A file named 'Sample Data.zip' is listed, with a yellow box highlighting it. The file was added by 'chriskollhed' and has a size of 782 KB. A yellow box also highlights the 'Download' button. At the bottom of the page, there's a link to the raw file: github.com/SAP-samples/hana-cloud-learning/raw/_/Sample%20Data.zip.

- You need to **unzip** the Sample Data.zip file before it can be used to upload into the table.

3. Go back to the Database Explorer, right-click on the **INVENTORY** table and select **Import Data**.



4. A wizard opens where you can select **Import Data** as IMPORT TYPE. Click on **Step 2**.
 5. Under IMPORT SOURCE, select **Local** to specify where the data is imported from. Click on **Browse** to the find the unzipped .csv file on your local machine. Click on the **.csv** file and select open. Click on **Step 3**.



6. Under **IMPORTTARGET**, name the Table as **INVENTORY** unless it's not named so by default. Click on **Step 4**.

7. Under **TABLE MAPPING**, verify that the column properties match with the column definition that you had used to create the **inventory.hdbtable**. After the check is complete, click on **Step 5**.

Source Column ...	Use Default Value	Database Column...	Data Type	Length	Scale	Key	Not Null
SOURCE...	<input type="checkbox"/>	SOURCEREGION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
SOURCE...	<input type="checkbox"/>	SOURCEREGIODESCRIPTION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
SOURCE...	<input type="checkbox"/>	SOURCESITE	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
SOURCE...	<input type="checkbox"/>	SOURCESITESITESDESCRIPTION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
SOURCE...	<input type="checkbox"/>	SOURCELAT	DECIMAL	10	6	<input type="checkbox"/>	<input type="checkbox"/>
SOURCE...	<input type="checkbox"/>	SOURCELONG	DECIMAL	10	6	<input type="checkbox"/>	<input type="checkbox"/>
LOGISTIC...	<input type="checkbox"/>	LOGISTICCENTER	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
LOGISTIC...	<input type="checkbox"/>	LOGISTICCENTERDESCRIPTION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
PHYSICA...	<input type="checkbox"/>	PHYSICALROUTE	NVARCHAR	1		<input type="checkbox"/>	<input type="checkbox"/>
PHYSICA...	<input type="checkbox"/>	PHYSICALROUTECONNECTIONDESCRIPTION	NVARCHAR	1		<input type="checkbox"/>	<input type="checkbox"/>

Step 5

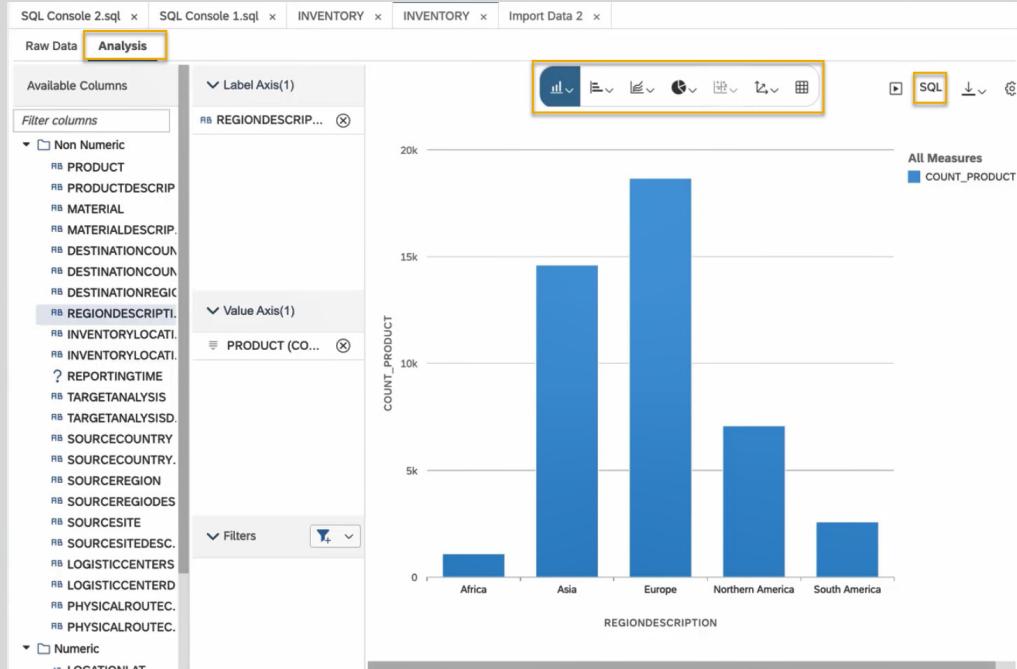
8. Under ERROR HANDLING, select the option **Save all successful rows and list the errors (if any)**. Click on **Review**.
9. After reviewing the Import Summary, click **Import Into Database**. Wait until the Import Status shows the **Import complete** notification.

Source Column...	Use Default Val...	Database Colu...	Data Type	Length	Scale	Key	Not Null
PRODUCT	<input type="checkbox"/>	PRODUCT	NVARCHAR	10		<input type="checkbox"/>	<input type="checkbox"/>
PRODUCTDESCRIPTION	<input type="checkbox"/>	PRODUCTDESCRIPTION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
MATERIAL	<input type="checkbox"/>	MATERIAL	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
MATERIALDESCRIPTION	<input type="checkbox"/>	MATERIALDESCRIPTION	NVARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>
DESTINATIONCOUNTRY	<input type="checkbox"/>	DESTINATIONCOUNTRY	NVARCHAR	2		<input type="checkbox"/>	<input type="checkbox"/>
DESTINATIONCO	<input type="checkbox"/>	DESTINATIONCO					

10. To verify the data import, click on **INVENTORY** table from the left side panel. Select **Open Data** to view the data added into the table.

Sidenote: Creating graphs in the SAP HANA Database Explorer

- Now that your data is imported successfully, you can preview it using the Analysis functions of the SAP HANA Database Explorer.
- In the Open Data window, click on **Analysis** to view the data in the form of graphs.
- For example, you can drag **PRODUCT** from the list of Available Columns to the **Value Axis** and **REGIONDESCRIPTION** to the **Label Axis** to see how products are distributed over regions.



- You can select different graphical options provided in a main toolbar at the top or access the corresponding SQL code to these graphical views.

3. CHANGE DATA STRUCTURES USING HDBTABLE

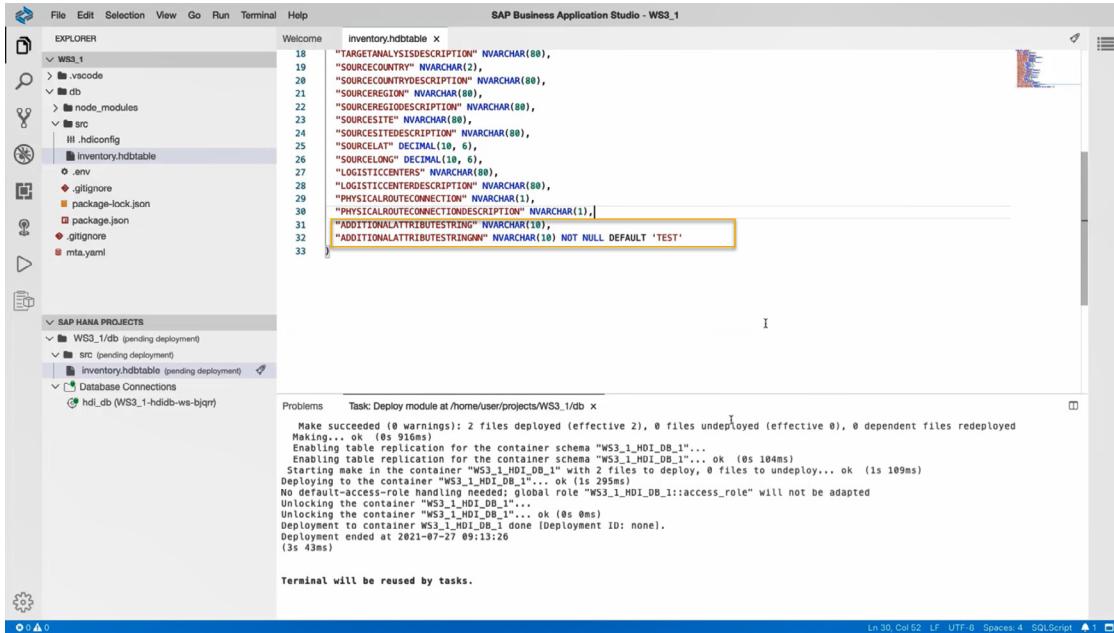
Now that your table is created and the data is imported, we will change the structure of this table by adding two columns. You can then examine in the terminal how this type of object handles data structure changes, which we will compare to other table types later on.

1. Go back to SAP Business Application Studio and select your **inventory.hdbtable**.
2. Add two columns to the table definition by copying the following code and pasting at the end of the **line 30**.

```
"ADDITIONALATTRIBUTESTRING" NVARCHAR(10),
"ADDITIONALATTRIBUTESTRINGNN" NVARCHAR(10) NOT NULL DEFAULT 'TEST'
```

Make sure to paste these lines in between the last column definitions and the closing parenthesis ')' in **line 31**. The last previous line of column definitions needs to have a ';' at the end.

The new table definition should look like in this image:



3. Now you can deploy this updated object again to your database by clicking on the deploy icon in the SAP HANA Projects panel. You can either deploy only the file **inventory.hdbtable** or the whole project.
4. Once the table was successfully deployed, you can check if the data is still in the table by going back to the **SAP HANA Database Explorer** and **refreshing** the table.

5. In the table definitions, you should now see that two additional columns have been added to the table and opening the data should not result in any changes to the number of rows in the previous graph.

Table Name		Schema				
INVENTORY		WS3_1_HDI_DB_1				
Columns	Indexes	Properties	Runtime Information			
#	Name	SQL Data Type	Key	Not Null	Default	Comment
15	REPORTINGTIME	DATE			NULL	
16	TARGETANALYSIS	NVARCHAR(80)			NULL	
17	TARGETANALYSISDESCRIPTION	NVARCHAR(80)			NULL	
18	SOURCECOUNTRY	NVARCHAR(2)			NULL	
19	SOURCECOUNTRYDESCRIPTION	NVARCHAR(80)			NULL	
20	SOURCEREGION	NVARCHAR(80)			NULL	
21	SOURCEREGIODESCRIPTION	NVARCHAR(80)			NULL	
22	SOURCESITE	NVARCHAR(80)			NULL	
23	SOURCESITEDESCRIPTION	NVARCHAR(80)			NULL	
24	SOURCELAT	DECIMAL(10,6)			NULL	
25	SOURCELONG	DECIMAL(10,6)			NULL	
26	LOGISTICCENTERS	NVARCHAR(80)			NULL	
27	LOGISTICCENTERDESCRIPTION	NVARCHAR(80)			NULL	
28	PHYSICALROUTECONNECTION	NVARCHAR(1)			NULL	
29	PHYSICALROUTECONNECTIONDESCRIPTION	NVARCHAR(1)			NULL	
30	ADDITIONALATTRIBUTESTRING	NVARCHAR(10)			NULL	
31	ADDITIONALATTRIBUTESTRINGNN	NVARCHAR(10)	X	TEST		

You may also check the data in the table to verify that it contains the same data as it had earlier. You can do this by comparing the [output graph from the last section](#) to what you get with the current table.

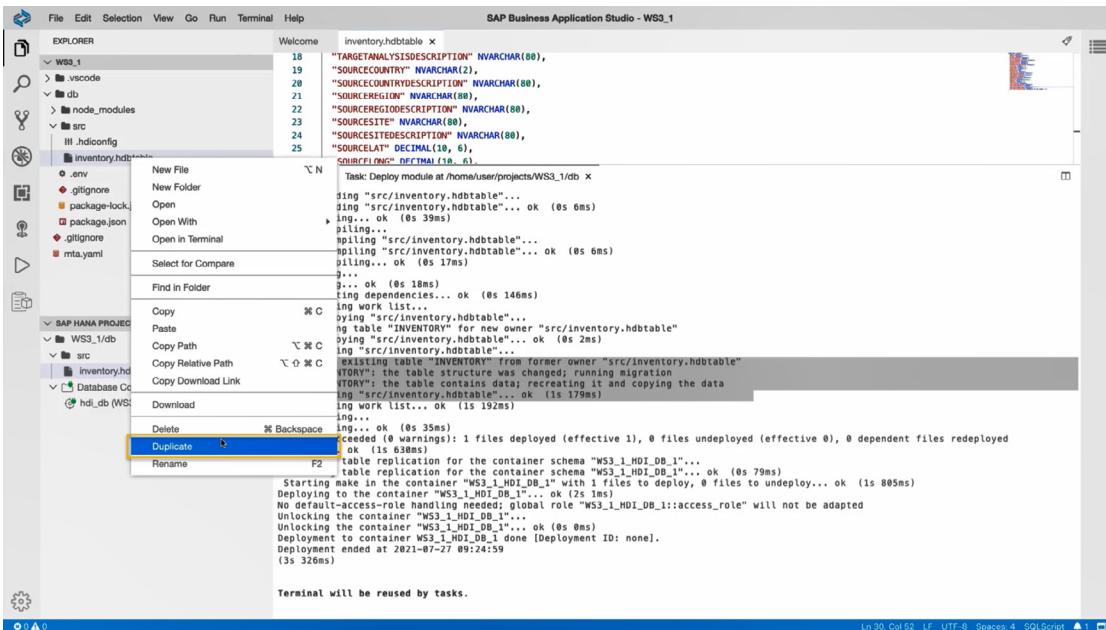
Note: While making changes to data structure using HDBTABLE, your database recreates the existing table with the updated table definition and copies the existing data into it. This is not the most efficient way to use your database capability if you have large amounts of data present in these tables. To highlight how to make changes to data structures efficiently, you will use an HDBMIGRATIONTABLE in the next section. In comparison to using an HDBTABLE, an HDBMIGRATIONTABLE can save considerable time and resources while handling changes in data structures.

4. CHANGE DATA STRUCTURES USING HDBMIGRATIONTABLE

Now that you know how hdbtable objects handle data changes, you can use a different type of object to add two more rows to this table.

Recreate the table using an hdbmigrationtable object

1. Go back to SAP Business Application Studio. Right-click your **inventory.hdbtable** and select **Duplicate**.



2. Now, you can see a new file named **inventory_copy.hdbtable** in your src folder. Right-click on this file and select **Rename**. Rename the file as '**inventory.hdbmigrationtable**'.

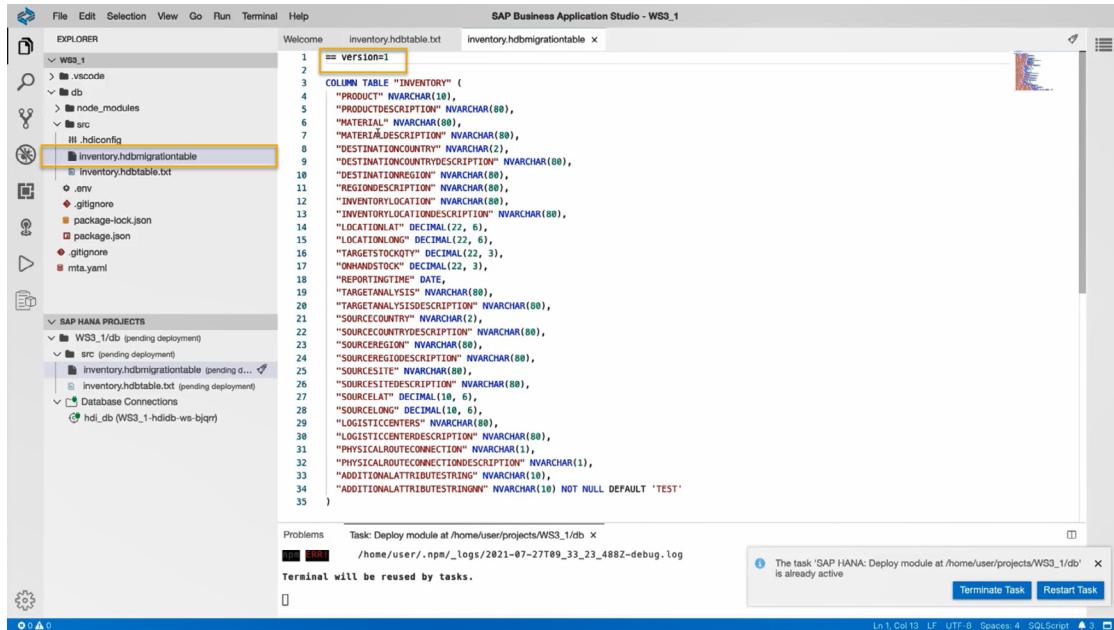
Important: Before you deploy the new file **inventory.hdbmigrationtable**, you must make a change to the already existing **inventory.hdbtable** file. Otherwise, the deployment will result in an error. Adding the ending '.txt' to the file name will signal to SAP Business Application Studio that this file shall not be deployed to create an HDI object in the database.

3. Rename the **inventory.hdbtable** as **inventory.hdbtable.txt** to facilitate the deployment of the new **inventory.hdbmigrationtable** object.

4. Now, click on **inventory.hdbmigrationtable** file. Copy and paste the following code at the beginning of the file such that it comes before the table definition.

```
== version = 1
```

You can check the image given below.



```

== version = 1
1 COLUMN TABLE "INVENTORY" (
2   "PRODUCT" NVARCHAR(10),
3   "HANID" NVARCHAR(100),
4   "MATERIAL" NVARCHAR(100),
5   "MATERIALDESCRIPTION" NVARCHAR(80),
6   "DESTINATIONCOUNTRY" NVARCHAR(1),
7   "DESTINATIONCOUNTRYDESCRIPTION" NVARCHAR(80),
8   "DESTINATIONREGION" NVARCHAR(80),
9   "REGIONDESCRIPTION" NVARCHAR(80),
10  "INVENTORYLOCATION" NVARCHAR(80),
11  "INVENTORYLOCATIONDESCRIPTION" NVARCHAR(80),
12  "LOCATIONLAT" DECIMAL(22, 6),
13  "LOCATIONLONG" DECIMAL(22, 6),
14  "TARGETSTOCKQTY" DECIMAL(22, 3),
15  "ONHANDSTOCK" DECIMAL(22, 3),
16  "REPORTINGTIME" DATE,
17  "TARGETANALYSIS" NVARCHAR(80),
18  "SOURCECOUNTRY" NVARCHAR(100),
19  "SOURCECOUNTRYDESCRIPTION" NVARCHAR(80),
20  "SOURCEREGION" NVARCHAR(80),
21  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
22  "SOURCEREGIODESCRIPTION" NVARCHAR(80),
23  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
24  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
25  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
26  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
27  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),
28  "SOURCELAT" DECIMAL(10, 6),
29  "SOURCELONG" DECIMAL(10, 6),
30  "LOGISTICCENTERS" NVARCHAR(80),
31  "LOGISTICCENTERDESCRIPTION" NVARCHAR(80),
32  "PHYSICALROUTECONNECTION" NVARCHAR(1),
33  "PHYSICALROUTECONNECTIONDESCRIPTION" NVARCHAR(1),
34  "ADDITIONALATTRIBUTESTRING" NVARCHAR(10),
35  "ADDITIONALATTRIBUTESTRINGN" NVARCHAR(10) NOT NULL DEFAULT 'TEST'
)

```

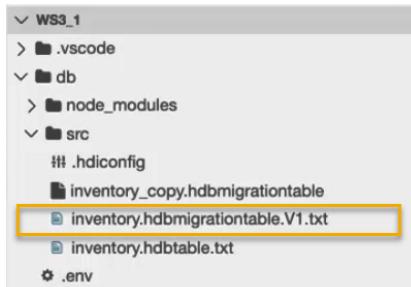
The screenshot shows the SAP Business Application Studio interface. The left sidebar displays the SAP HANA Projects panel, which includes a list of files and pending deployments. In the center, the code editor shows the migration script for the 'INVENTORY' table. A yellow box highlights the first line of the script: '== version = 1'. Below the code editor, a terminal window shows deployment logs, indicating a task is already active. The bottom status bar shows standard developer information like line count, character count, and encoding.

5. On the bottom left corner, you can see the file **inventory.hdbmigrationtable** in the SAP HANA Projects panel. Click on the  deploy icon for the file to deploy the changes to your HDI container.
6. In the terminal you can see that the table was successfully deployed.
7. Go back to the **SAP HANA Database Explorer**, right-click on **Tables** and select **Refresh** to update the changes you made to the table.
8. Click on the **INVENTORY** table to view the list of all the columns present. Here, you can verify that deploying the inventory.hdbmigrationtable has produced exactly the same results as with inventory.hdbtable.

Add rows to the hdbmigrationtable object

The next step is to change the data structure of the table by adding two new columns. We will again duplicate our table change the previous version of this file to a .txt file that will not be deployed. Then, we will adjust the code of the hdbmigration object to define how to migrate from Version 1 of hdbmigrationtable to Version 2. For this, follow the next steps.

1. Go back to SAP Business Application Studio.
2. In the EXPLORER panel, right-click on the **inventory.hdbmigrationtable** and select **Duplicate**.
3. Now, you can see a new file named **inventory_copy.hdbmigrationtable** in your src folder.
4. Right-click on the original **inventory.hdbmigration** file and rename this file as '**inventory.hdbmigrationtable.V1.txt**' to facilitate the deployment of the new copy of the hdbmigrationtable.



5. Next, right-click on the **inventory_copy.hdbmigrationtable** file and select **Rename**. Rename this file as **inventory.hdbmigrationtable**.
6. Now, click on the **inventory.hdbmigrationtable** file. Adjust the first line of code to version 2 and paste the following two lines of code at the end of the file.v

```
"ADDINTEGER" INTEGER,  
"ADDINTEGER_NN" INTEGER NOT NULL DEFAULT 0
```

You can check the image given below.

```
1 == version=2  
2  
3 COLUMN TABLE "INVENTORY" {  
4   "PRODUCT" NVARCHAR(10),  
5   "PRODUCTDESCRIPTION" NVARCHAR(80),  
6   "MATERIAL" NVARCHAR(80),  
7   "MATERIALDESCRIPTION" NVARCHAR(80),  
8   "DESTINATIONCOUNTRY" NVARCHAR(2),  
9   "DESTINATIONCOUNTRYDESCRIPTION" NVARCHAR(80),  
10  "DESTINATIONREGION" NVARCHAR(80),  
11  "REGIONDESCRIPTION" NVARCHAR(80),  
12  "INVENTORYLOCATION" NVARCHAR(80),  
13  "INVENTORYLOCATIONDESCRIPTION" NVARCHAR(80),  
14  "LOCATIONLAT" DECIMAL(22, 6),  
15  "LOCATIONLONG" DECIMAL(22, 6),  
16  "TARGETSTOCKOTY" DECIMAL(22, 3),  
17  "ONHANDSTOCK" DECIMAL(22, 3),  
18  "REPORTINGTIME" DATE,  
19  "TARGETANALYSIS" NVARCHAR(80),  
20  "TARGETANALYSISDESCRIPTION" NVARCHAR(80),  
21  "SOURCECOUNTRY" NVARCHAR(2),  
22  "SOURCECOUNTRYDESCRIPTION" NVARCHAR(80),  
23  "SOURCEREGION" NVARCHAR(80),  
24  "SOURCEREGIONDESCRIPTION" NVARCHAR(80),  
25  "SOURCESITE" NVARCHAR(80),  
26  "SOURCESITEDESCRIPTION" NVARCHAR(80),  
27  "SOURCELAT" DECIMAL(10, 6),  
28  "SOURCELONG" DECIMAL(10, 6),  
29  "LOGISTICCENTERS" NVARCHAR(80),  
30  "LOGISTICCENTERDESCRIPTION" NVARCHAR(80),  
31  "PHYSICALROUTECONNECTION" NVARCHAR(1),  
32  "PHYSICALROUTECONNECTIONDESCRIPTION" NVARCHAR(1),  
33  "ADDITIONALATTRIBUTESTRING" NVARCHAR(10),  
34  "ADDITIONALATTRIBUTESTRINGNN" NVARCHAR(10) NOT NULL DEFAULT 'TEST',  
35  "ADDINTEGER" INTEGER,  
36  "ADDINTEGER_NN" INTEGER NOT NULL DEFAULT 0
```

7. An hdbmigrationtable object also needs to have a block of code at the very end that defines all migration steps. The following code will do just that, so paste it below the closing parenthesis of the table definition.

```

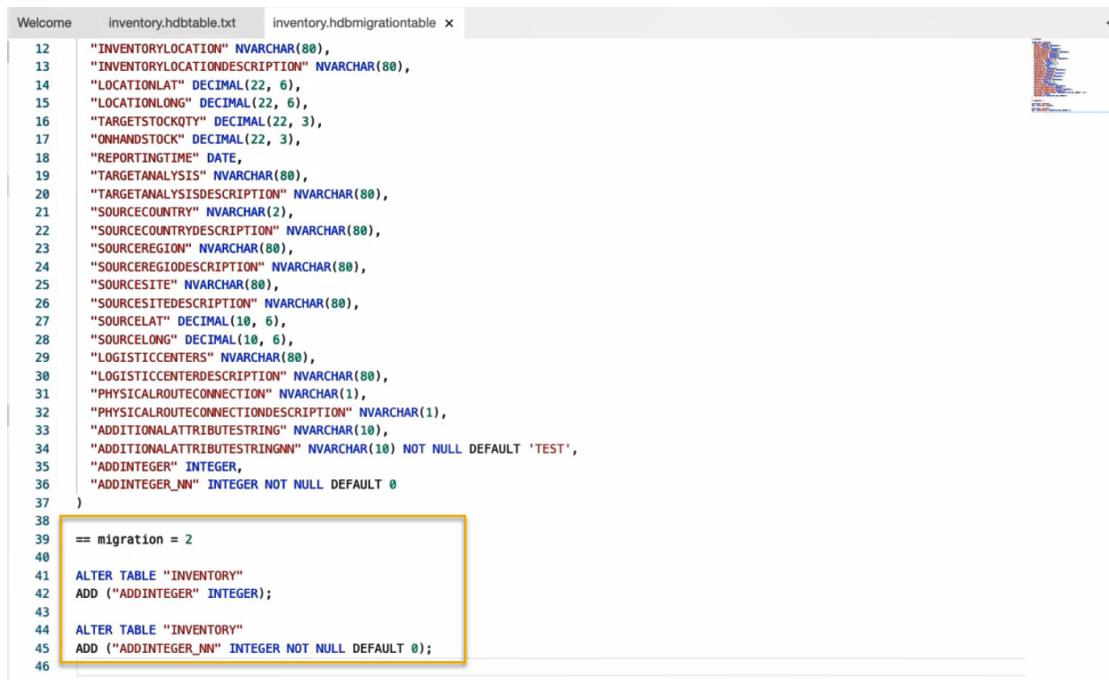
== migration = 2

ALTER TABLE "INVENTORY"
ADD ("ADDINTEGER" INTEGER);

ALTER TABLE "INVENTORY"
ADD ("ADDINTEGER_NN" INTEGER NOT NULL DEFAULT 0);

```

You can check the image given below.



The screenshot shows a code editor window in SAP Studio. The file is named 'inventory.hdbmigrationtable'. The code defines a table 'INVENTORY' with various columns and then adds two new columns 'ADDINTEGER' and 'ADDINTEGER_NN' using the ALTER TABLE command. The last two lines of code, which define the migration steps, are highlighted with a yellow box.

```

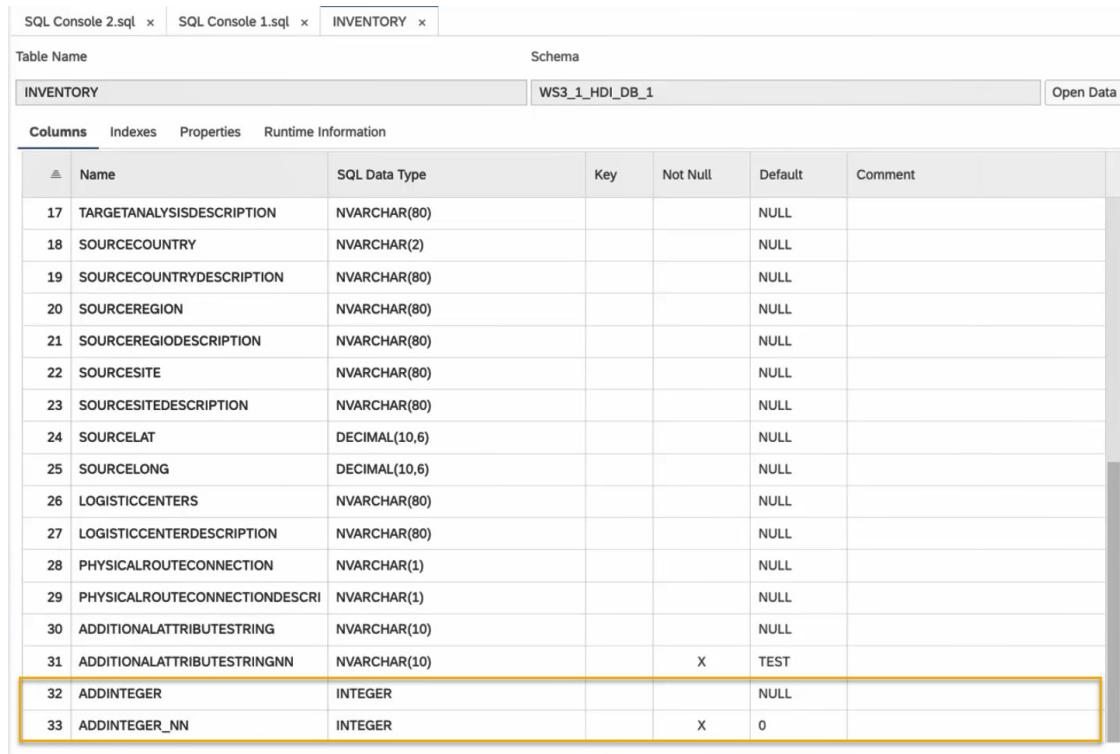
12   "INVENTORYLOCATION" NVARCHAR(80),
13   "INVENTORYLOCATIONDESCRIPTION" NVARCHAR(80),
14   "LOCATIONLAT" DECIMAL(22, 6),
15   "LOCATIONLONG" DECIMAL(22, 6),
16   "TARGETSTOCKQTY" DECIMAL(22, 3),
17   "ONHANDSTOCK" DECIMAL(22, 3),
18   "REPORTINGTIME" DATE,
19   "TARGETANALYSIS" NVARCHAR(80),
20   "TARGETANALYSISDESCRIPTION" NVARCHAR(80),
21   "SOURCECOUNTRY" NVARCHAR(2),
22   "SOURCECOUNTRYDESCRIPTION" NVARCHAR(80),
23   "SOURCEREGION" NVARCHAR(80),
24   "SOURCEREGIODESCRIPTION" NVARCHAR(80),
25   "SOURCESITE" NVARCHAR(80),
26   "SOURCESITEDESCRIPTION" NVARCHAR(80),
27   "SOURCELAT" DECIMAL(10, 6),
28   "SOURCELONG" DECIMAL(10, 6),
29   "LOGISTICCENTERS" NVARCHAR(80),
30   "LOGISTICCENTERDESCRIPTION" NVARCHAR(80),
31   "PHYSICALROUTECONNECTION" NVARCHAR(1),
32   "PHYSICALROUTECONNECTIONDESCRIPTION" NVARCHAR(1),
33   "ADDITIONALATTRIBUTESTRING" NVARCHAR(10),
34   "ADDITIONALATTRIBUTESTRINGNN" NVARCHAR(10) NOT NULL DEFAULT 'TEST',
35   "ADDINTEGER" INTEGER,
36   "ADDINTEGER_NN" INTEGER NOT NULL DEFAULT 0
37 )
38 == migration = 2
39
40
41 ALTER TABLE "INVENTORY"
42 ADD ("ADDINTEGER" INTEGER);
43
44 ALTER TABLE "INVENTORY"
45 ADD ("ADDINTEGER_NN" INTEGER NOT NULL DEFAULT 0);

```

Note: The above step changes the data structure by adding two new columns into the table by using the ALTER command. This way you can efficiently avoid the need to recreate the table and copy data into it.

8. Now, deploy your project to the database by clicking on the  deploy icon on the project level **WS3_1** **db** In the terminal you can see that the table was successfully deployed. Follow the next steps to verify the changes made to the column definition.
9. Go back to the **SAP HANA Database Explorer** and **Refresh** your table to update the changes made to the table.

10. Click on the **INVENTORY** table to view the list of all the columns present. Here, you can see that two additional columns have been added to the table.



The screenshot shows the SAP SQL Console interface with three tabs at the top: SQL Console 2.sql, SQL Console 1.sql, and INVENTORY. The INVENTORY tab is active. Below the tabs, the title "INVENTORY" is shown under "Table Name" and "Schema WS3_1_HDI_DB_1". A "Open Data" button is also visible. The main area displays the table schema with 33 columns. The columns are listed in rows, with their names, data types, and various properties like Key, Not Null, Default, and Comment. The last two columns, ADDINTEGER and ADDINTEGER_NN, are highlighted with a yellow border.

	Name	SQL Data Type	Key	Not Null	Default	Comment
17	TARGETANALYSISDESCRIPTION	NVARCHAR(80)			NULL	
18	SOURCECOUNTRY	NVARCHAR(2)			NULL	
19	SOURCECOUNTRYDESCRIPTION	NVARCHAR(80)			NULL	
20	SOURCEREGION	NVARCHAR(80)			NULL	
21	SOURCEREGIODESCRIPTION	NVARCHAR(80)			NULL	
22	SOURCESITE	NVARCHAR(80)			NULL	
23	SOURCESITEDESCRIPTION	NVARCHAR(80)			NULL	
24	SOURCELAT	DECIMAL(10,6)			NULL	
25	SOURCELONG	DECIMAL(10,6)			NULL	
26	LOGISTICCENTERS	NVARCHAR(80)			NULL	
27	LOGISTICCENTERDESCRIPTION	NVARCHAR(80)			NULL	
28	PHYSICALROUTECONNECTION	NVARCHAR(1)			NULL	
29	PHYSICALROUTECONNECTIONDESCR	NVARCHAR(1)			NULL	
30	ADDITIONALATTRIBUTESTRING	NVARCHAR(10)			NULL	
31	ADDITIONALATTRIBUTESTRINGNN	NVARCHAR(10)		X	TEST	
32	ADDINTEGER	INTEGER			NULL	
33	ADDINTEGER_NN	INTEGER		X	0	

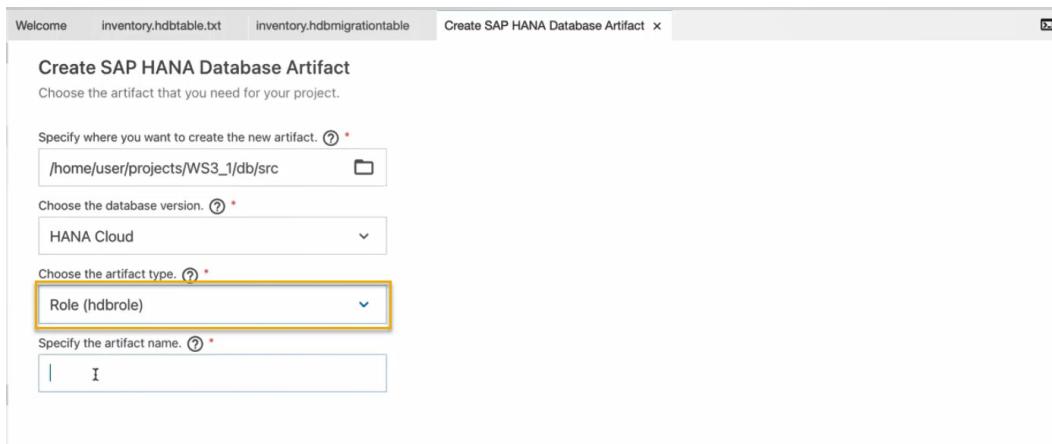
5. CREATE HDBROLES

Now that you know the difference between hdbtable and hdmigration table objects when changing data structures, we need to prepare some privileges for the upcoming workshop sessions. In the next section you will create roles with specific permissions to give other users access to these tables. This is necessary, e.g. for creating calculation views, which we will do in session 3.

Create a role with grant option

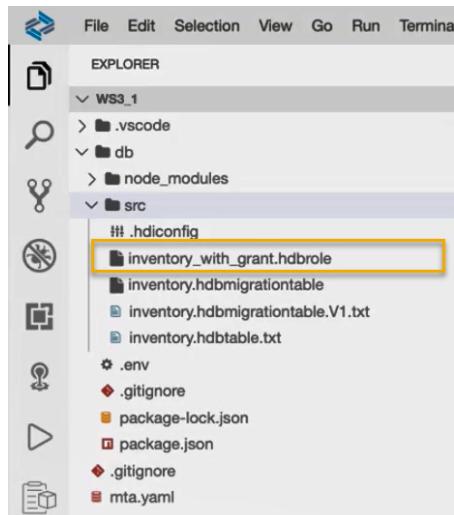
First, we want to create a role that contains SELECT privileges with grant option, which means when someone is assigned this role, they can grant this privilege to others themselves.

1. From the **EXPLORER** panel on the left side of SAP Business Application Studio, expand the folders under your **Project WS3_1**.
2. Select the **src folder** and press **F1 key** or click on **View – Find Command** at the top of the screen to open a search prompt. Enter **HANA** in the field and from the dropdown list of functionalities, choose **SAP HANA: Create SAP HANA Database Artifact**.
3. In the window for creating a new SAP HANA Database Artifact, click on the folder icon in the first field. Navigate to your project, the **src folder**, and then click on **Open**.
4. Select the artifact type as **Role (hdrole)**.



5. Under 'Specify the artifact name' option enter the name **inventory_with_grant**.
6. The database version should be HANA Cloud, by default. Without making any further changes, click on **Create**.

7. A new file **inventory_with_grant.hdbrole** can be seen under the src folder in the EXPLORER panel. Click on this role file to open the Role Editor window.

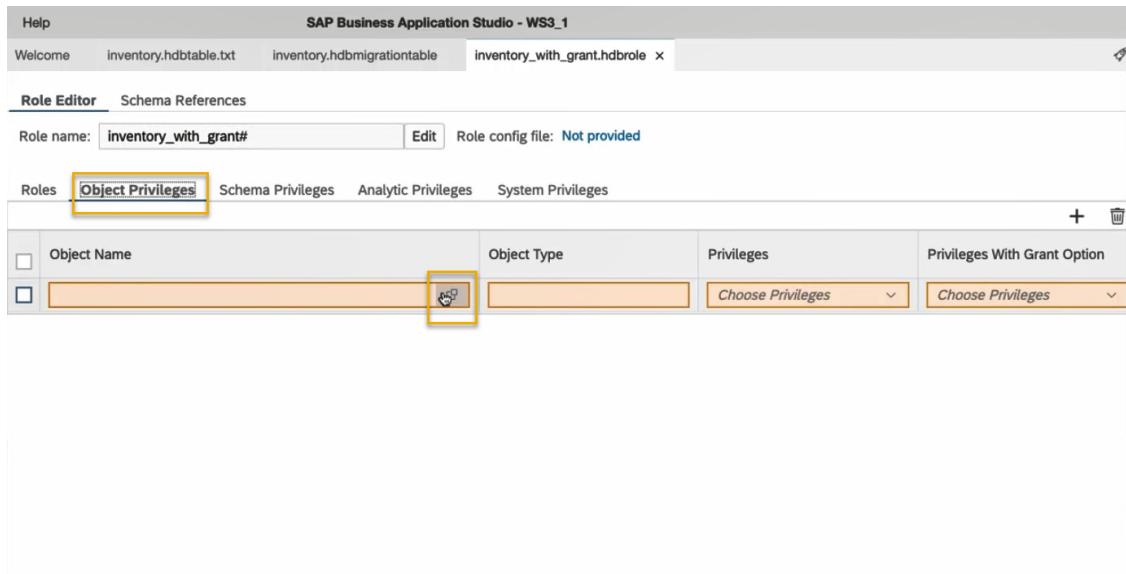


8. Click on Edit for the Role name. Rename it as **inventory_with_grant#**.

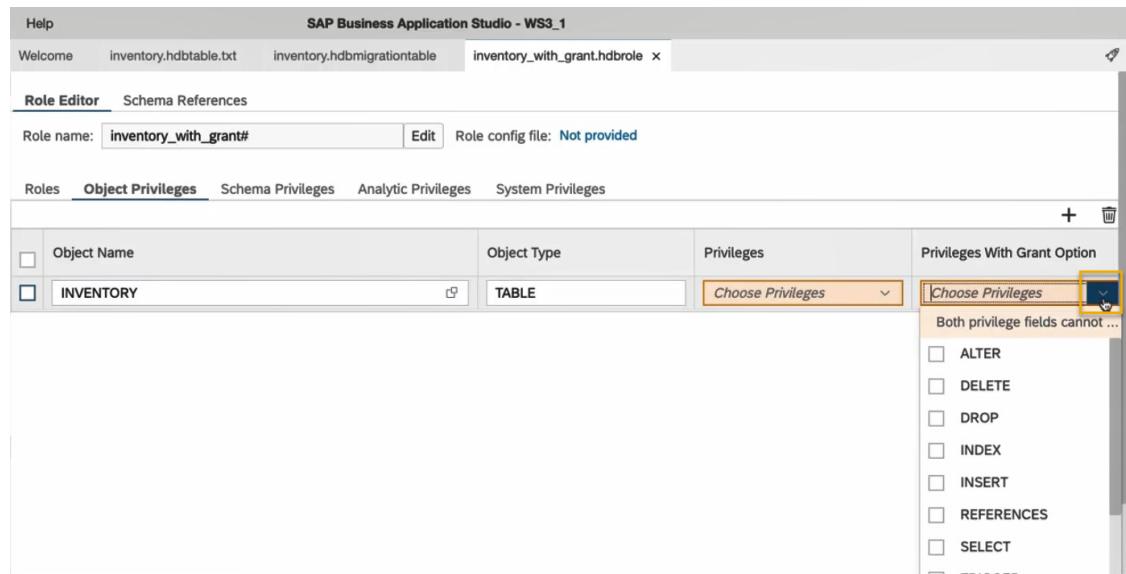
The screenshot shows the SAP Business Application Studio interface with the Role Editor tab selected. The title bar says 'SAP Business Application Studio - WS3_1'. The 'Role name:' field contains 'inventory_with_grant#' (highlighted with a yellow box). Below the table, there is a note: 'Role config file: Not provided'. The table has columns for 'Role Name' and 'Schema Reference'. There is one row where the 'Role Name' column is empty and the 'Schema Reference' column contains a placeholder 'Select schema reference...'. A 'Manage schema references' button is also visible.

Important: The '#' character at the end of the role name is necessary to create this role. Whenever you create roles with grant or admin options, they need to have a '#' character at the end of the role name.

9. Next, click on the tab **Object Privileges**. In the column **Object Name**, click on the  icon in the first row.



10. A wizard will open with all the objects available to you to select from. Since we only created one object so far, the only object available is the table **INVENTORY**. Click on it to select it.
11. In the column **Privileges with Grant Option**, click on the arrow where it says **Choose privileges** to select from a list of privileges with grant option that you want to assign to this role.



Object Name	Object Type	Privileges	Privileges With Grant Option
INVENTORY	TABLE	Choose Privileges	Choose Privileges

Both privilege fields cannot ...

- ALTER
- DELETE
- DROP
- INDEX
- INSERT
- REFERENCES
- SELECT
- TRIGGER

12. Check the **Select** privilege.

Create a second role without grant option

Now you will create a new role having privileges without the grant option. Repeat the steps from the previous section with a few modifications:

1. When creating the object, under 'Specify the artifact name' enter the name **inventory**.
2. Once the file **inventory.hdbrole** can be seen under the src folder in the EXPLORER panel, open this role file in the Role Editor.
3. Again, in the tab **Object Privileges**, select the table INVENTORY in the field object privileges.
4. In the column **Privileges**, click on the arrow where it says **Choose privileges** to select from a list of privileges that you want to assign to this role.

5. Check the **Select** privilege.

Deploy the roles to your database

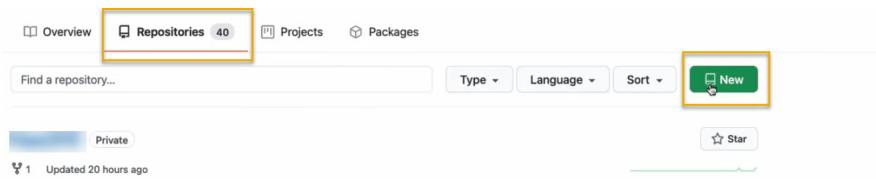
Now that both roles are created, you can deploy both files individually or deploy the whole project by clicking on the  deploy icon in the SAP HANA Projects panel. Once you see in the terminal that the project was deployed successfully, you can continue.

Note: These roles will be used in Session 3 to share calculation views among different users.

SYNCHRONIZE YOUR PROJECT WITH GIT

When collaborating with others on development projects, pushing your files to a git repository is very helpful. In this last section, we will establish a connection to GitHub, create a new repository and sync our development project with the repository.

1. Open github.com on a new browser tab. Enter your GitHub account using your credentials.
2. On the top right corner of the screen, click on the profile icon in the top right corner and select **Your repositories**.
3. In the top right corner, click on **New** to create a new repository.



4. Under Create a new repository, enter **WS3_1** as the Repository name.
5. Optionally, you can add a short description.
6. You can choose to make your repository visible publicly or privately. We suggest you select the option **Private**.
7. Click on **Create repository** to finish. Your first step is done and you have an empty repository ready to have the content from your development project.

Repository template
Start your repository with a template repository's contents.
[No template](#)

Owner * **Repository name ***

/ WS3_1 ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-fortnight](#)?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

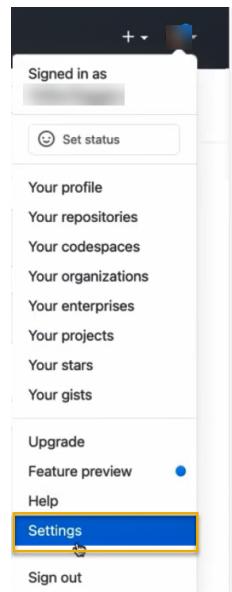
Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

Create a GitHub token

There are two ways of connecting to a github project: SSH and HTTPS. SSH requires you to use your github password and HTTPS requires the use of a token. We recommend using HTTPS for higher security. In this step we will create a github token that can be used to connect to git.

1. On the top right corner of the screen, click on your GitHub profile icon and choose **Settings**.



2. Scroll down the **Settings** page to find **Developer settings** on the left side of the screen and select it.
3. In **Developer settings**, select **Personal access tokens**.
4. Click on **Generate new token** at the top right corner of the screen. You will be asked to sign in using your GitHub account password.

A screenshot of the GitHub Personal access tokens page. At the top, there are navigation links: Issues, Marketplace, and Explore. Below them is a header with 'Personal access tokens' and buttons for 'Generate new token' (highlighted with a yellow box) and 'Revoke all'. A note says 'Tokens you have generated that can be used to access the GitHub API.' Below this is a table with one row. The row shows a placeholder image for a token, the text 'Last used within the last week', and a 'Delete' button. A note at the bottom left says '⚠ This token has no expiration date.'.

5. After signing in, **enter a name for the token** (for example: MyNewToken) in the field **Note**.

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over Basic Authentication.

Note

MyNewToken

Expiration *

30 days The token will expire on Thu, Aug 26 2021

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

6. Next you can select an expiration date. It's up to you what timeframe you want to select here.
7. Under **Select scopes**, you can select both **repo** and **notifications**. Click on **Generate token** to complete the generation of the token.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

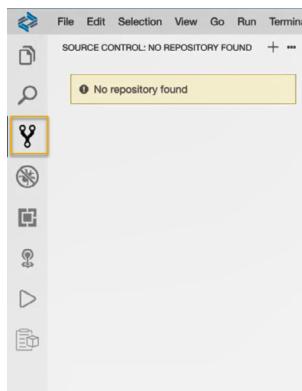
- Copy the new token you generated and paste it in a safe location that could be accessed later.

The screenshot shows the GitHub 'Personal access tokens' page. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' selected. The main area is titled 'Personal access tokens' and contains a message: 'Tokens you have generated that can be used to access the GitHub API.' Below this is a note: 'Make sure to copy your personal access token now. You won't be able to see it again!' A newly generated token is highlighted with a yellow box. This token has a checkmark icon, a copy icon, and a delete button. A tooltip above the token says: 'This token has no expiration date.' Other tokens listed below it also have similar status messages: 'Regenerate this token to take advantage of the new token formats' and 'This token has no expiration date.' Each token entry includes a 'Delete' button.

Important: After you proceed to the next step, you cannot go back to view the token again unless you have saved it in another location. In such a case, you will need to generate a new token.

Connect your development project to GitHub

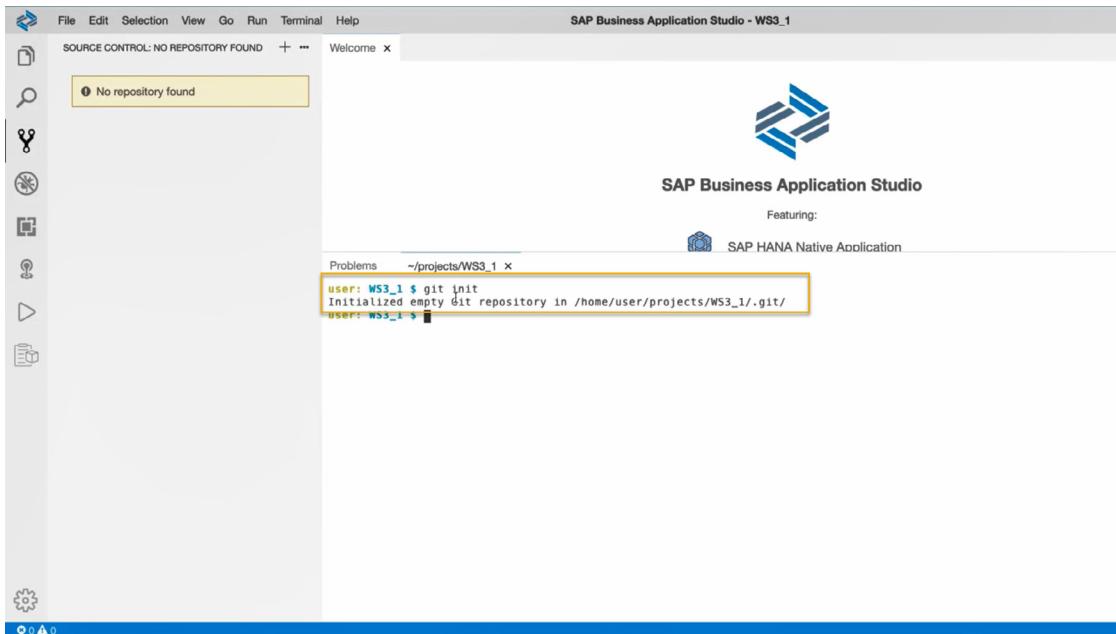
- Go back to SAP Business Application Studio.
- On the left menu, click on the icon to go to the **Source Control** panel. Here, you can see all the recent updates to your project that have not been pushed to GitHub yet. In this panel, you can also administer your GitHub connection and changes from a menu instead of using the terminal.



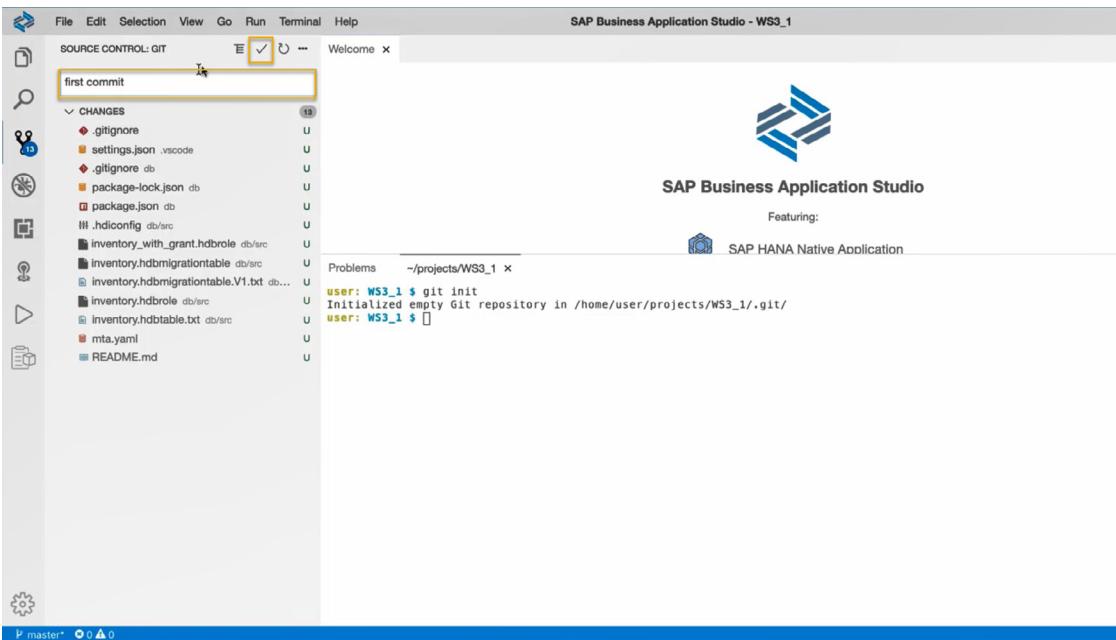
- Open a terminal by clicking on **Terminal** at the top navigation bar and select **Open Terminal**.

4. Paste the following command in the terminal and hit **Enter**. This will initialize the empty Git repository.

```
git init
```



5. Type **first commit** in the Message box under the Source Control panel. Click on the **tick mark** icon at the top of the panel to stage all your changes and commit them.



6. Depending on your preference, select **Yes** or **Always**, if you encounter a message box asking what action to take when the commit (tick mark) icon is selected.

You have now committed all the changes made to the data structures during this session. Next, you will add the remote origin and push the staged changes to your repository.

Add a remote origin via HTTPS connection

1. Go to the terminal and paste the following command. Before you hit Enter, replace <Username> with your **GitHub username** and <Repository name> with your **GitHub repository name**. (in this case for example: WS3_1)

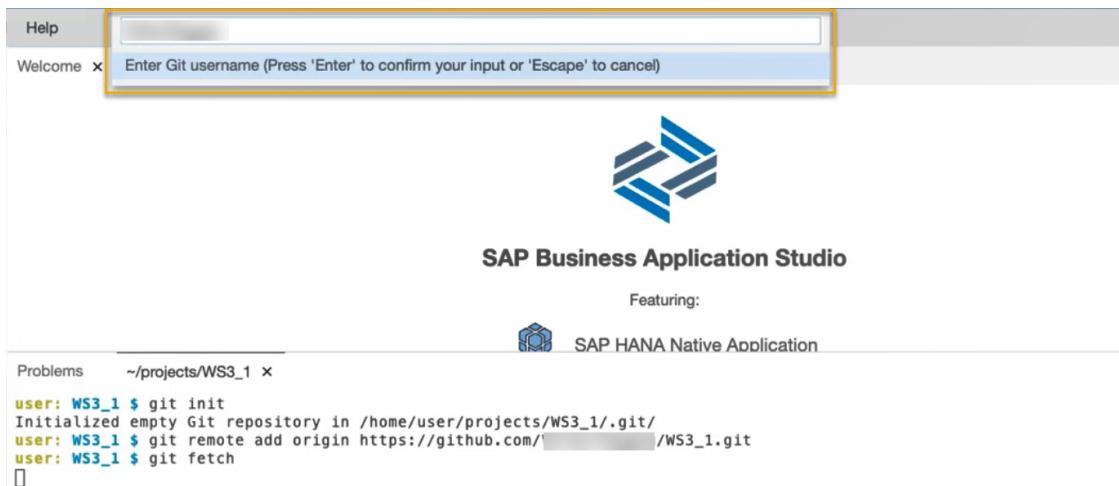
```
git remote add origin https://github.com/<Username>/<Repository name>.git
```

After making the changes, hit **Enter**.

2. Paste the following command into the terminal and hit **Enter**.

```
git fetch
```

3. A prompt will open at the top center of the screen. There, enter your Git username and press **Enter**.

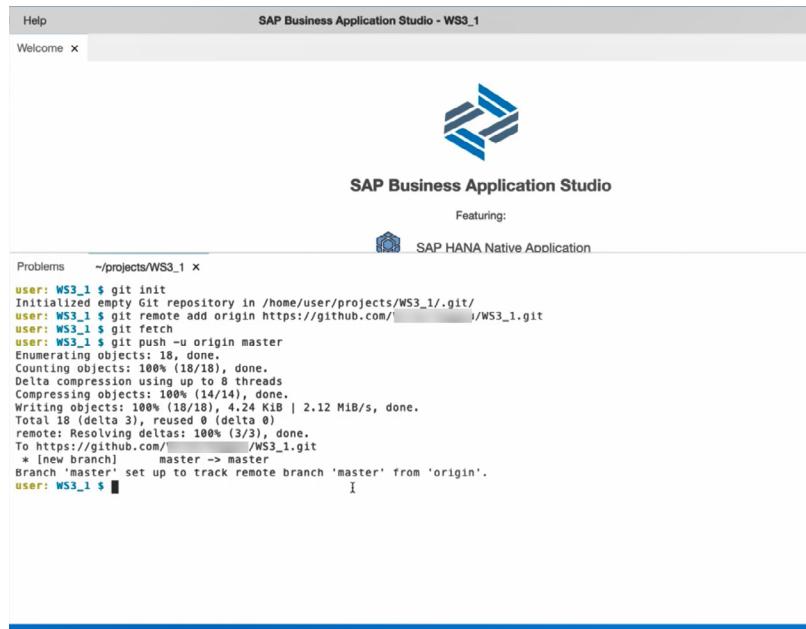


4. Next, paste the token you have generated from GitHub in the prompt and hit **Enter**.

5. You may save these credentials for future use by selecting the option to do so.

6. Paste the following command into the terminal and hit **Enter**.

```
git push -u origin master
```



SUMMARY

This pushes all the committed changes to the GitHub repository.

You can go back to your GitHub repository and verify that all the folders of your project are present in it.

And now you know how to manage changing data structures using HDBTABLE and HDBMIGRATIONTABLE design-time objects. In the next session, you will use CAP CDS to generate the design-time objects and automate the deployment process to the database.

Excellent! You've completed Session 1 of the Digital Hands-on Workshop!

Make sure to come back for session 2!

If you have any issues with the steps described in this workbook, please ask your questions in the [SAP Community](#) using the tag [SAP HANA Cloud](#).

Additional Resources

- [SAP HANA Cloud product page](#)
- [Upcoming SAP HANA Cloud events](#)
- [Explore SAP HANA Cloud tutorials](#)
 - [Mission: Jump start your SAP HANA Cloud, SAP HANA Database Trial](#)
 - [Mission: Get Started with a Standalone SAP HANA Cloud, Data Lake](#)
 - [Mission: Extend Your SAP HANA On-Premise to SAP HANA Cloud, SAP HANA Database](#)
- Technical Documentation
 - [SAP HANA Cloud](#)
 - [SAP Business Application Studio](#)
 - [SAP Continuous Integration and Delivery](#)