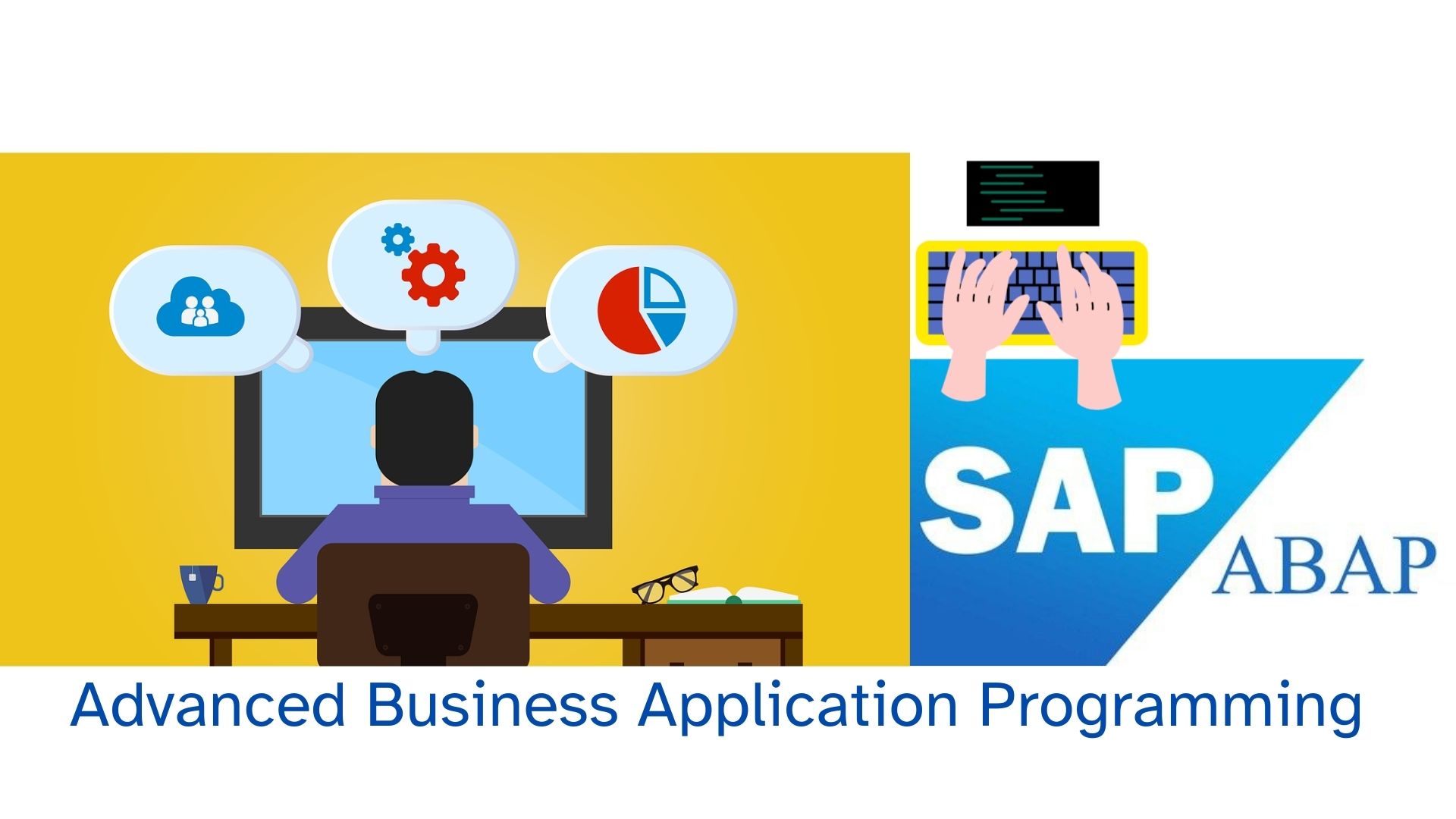
LAB MANUAL



Lab Manual

[Lab 1- Creating a BTP ABAP Environment 3](#_Toc145857416)

[Lab 2 - Creating an ABAP Package 10](#_Toc145857417)

[Lab 3 - Creating a Database Table 12](#_Toc145857418)

[Lab 4 - Create an ABAP Class 15](#_Toc145857419)

[Lab 5 - WAP to print Hello World 17](#_Toc145857420)

[Lab 6 - Perform Query Operation on the Output Table 18](#_Toc145857421)

[Lab 7 - Perform various ABAP data types of operation **Error! Bookmark not defined.**](#_Toc145857422)

[Lab 8 - Loops in ABAP, For loop, While Loop 23](#_Toc145857423)

[Lab 9 - Insert Data Table Entries and print them on the console 25](#_Toc145857424)

[Reference 27](#_Toc145857425)

Lab 1- Creating a BTP ABAP Environment

Create an SAP BTP ABAP Environment Trial User

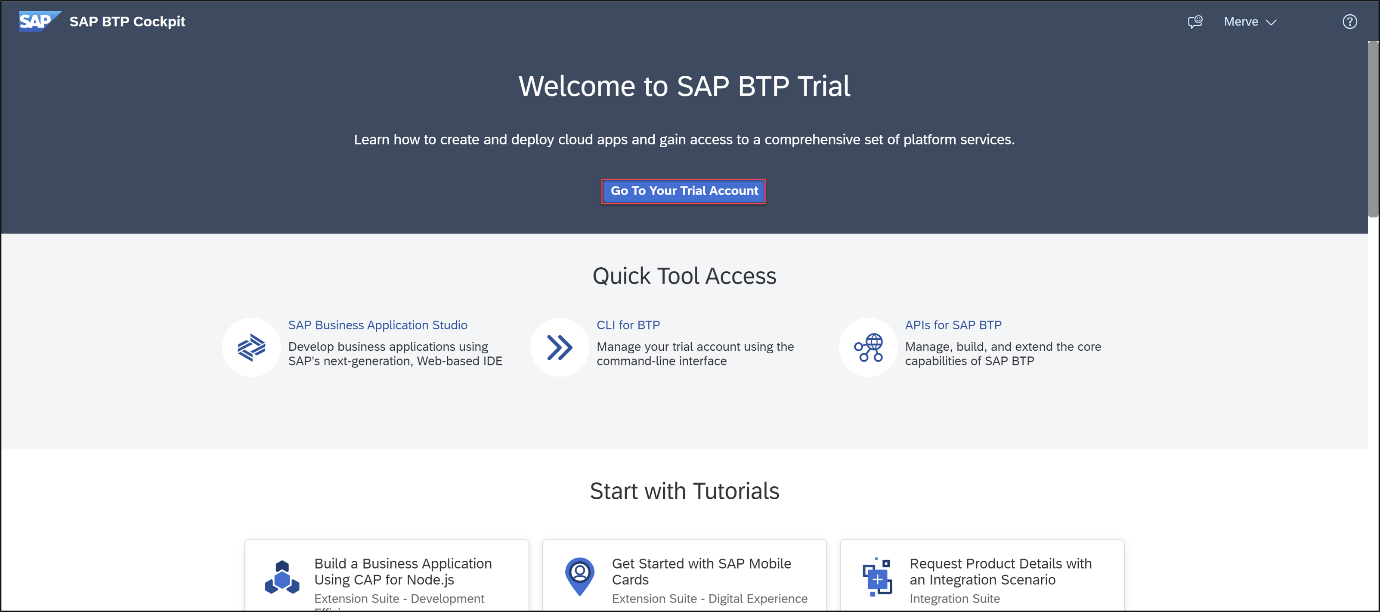
Create a trial user and ABAP cloud project with SAP BTP ABAP environment.

**We will learn**

* How to create a trial user
* How to create an ABAP Cloud project

Step 1: In your web browser, open the <https://cockpit.hanatrial.ondemand.com/>.

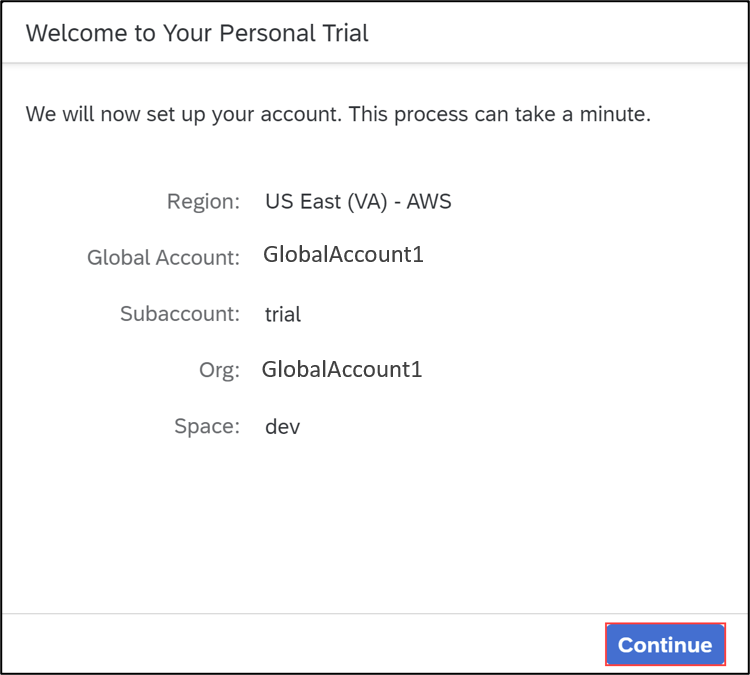
Navigate to the trial global account by clicking Go to Your Trial Account.



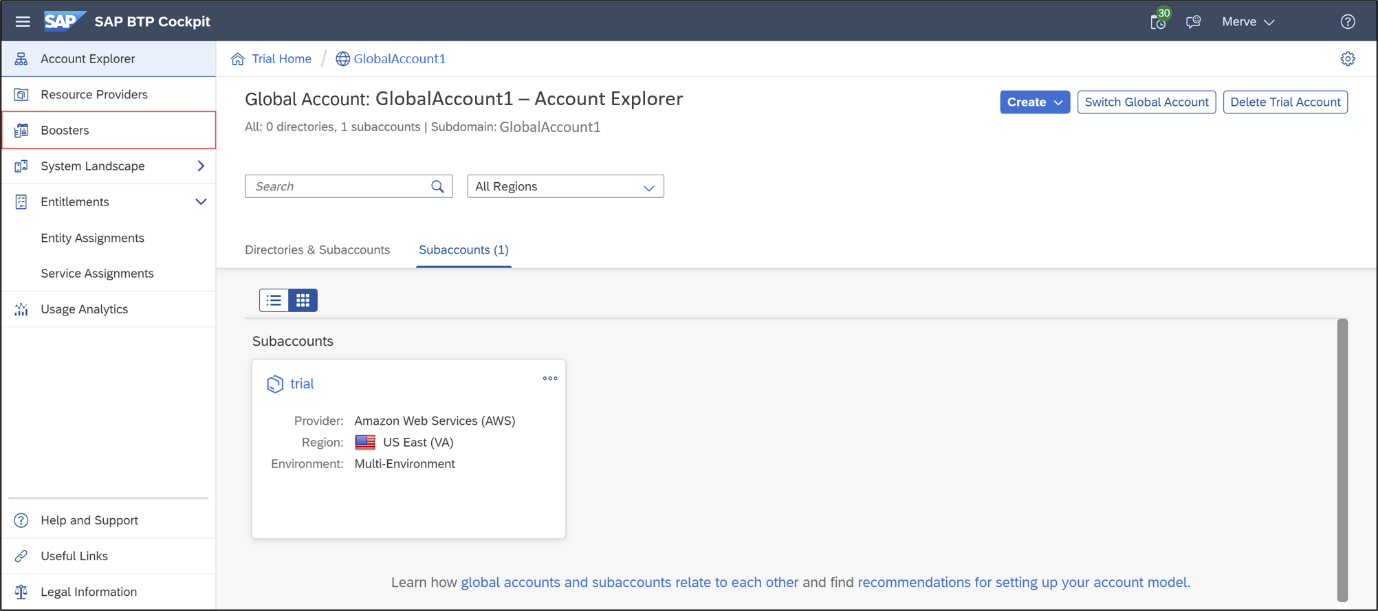
If this is your first time accessing your trial account, you’ll have to configure your account by choosing a region. Please select US East (VA) as a region. Your user profile will be set up for you automatically.

Wait till your account is set up and ready to go. Your global account, your subaccount, your organization, and your space are launched. This may take a couple of minutes.

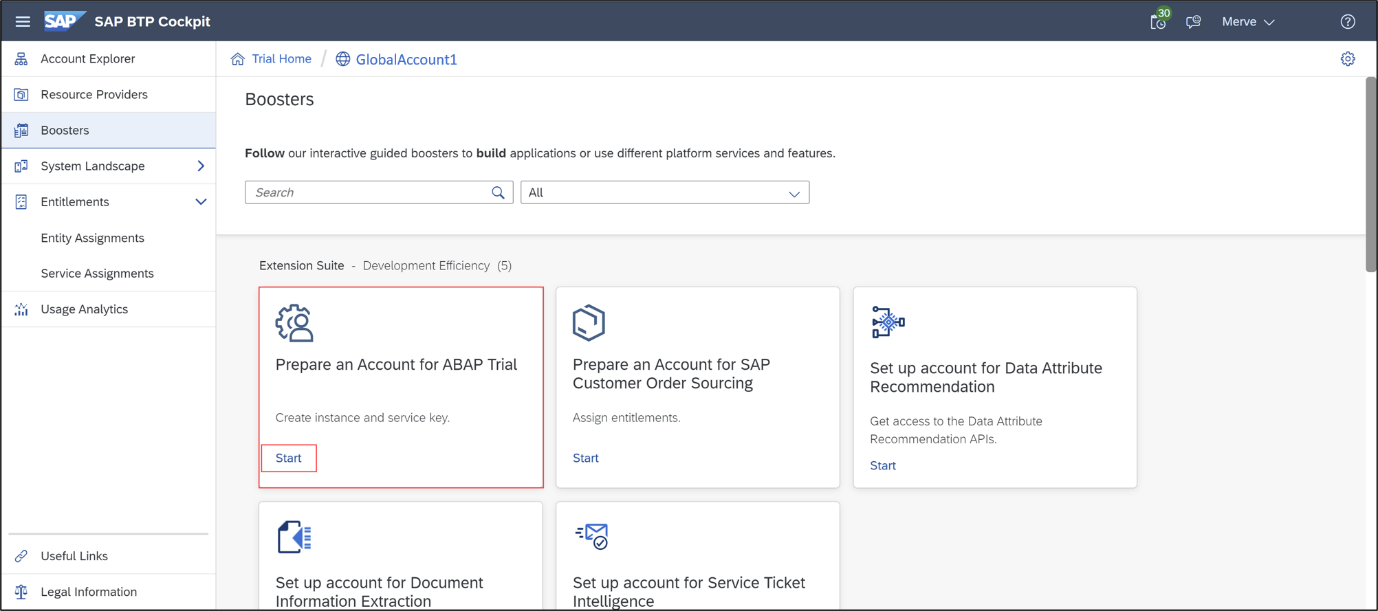
Choose Continue.



From your global account page, choose Boosters on the left side.



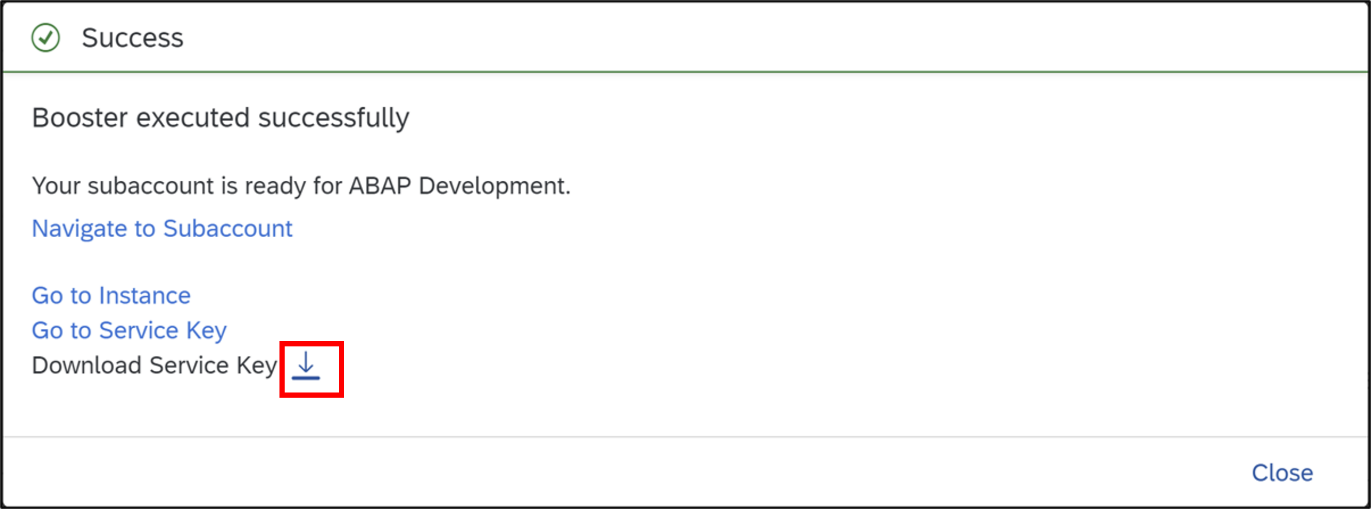
Search the Prepare an Account for ABAP Trial tile and press Start to start your booster. If you already created a service instance and service key, then please skip this step and move on with Step 2. Only one service instance can be created at a time.



Now the service instance and service key will be created for the ABAP trial user. The service key can be found inside the service instance.



The booster is now executed successfully. Download your service key for later use.



Step 2: Open ABAP Development Tools

**Open Eclipse. Make sure you have installed ADT in your Eclipse.**

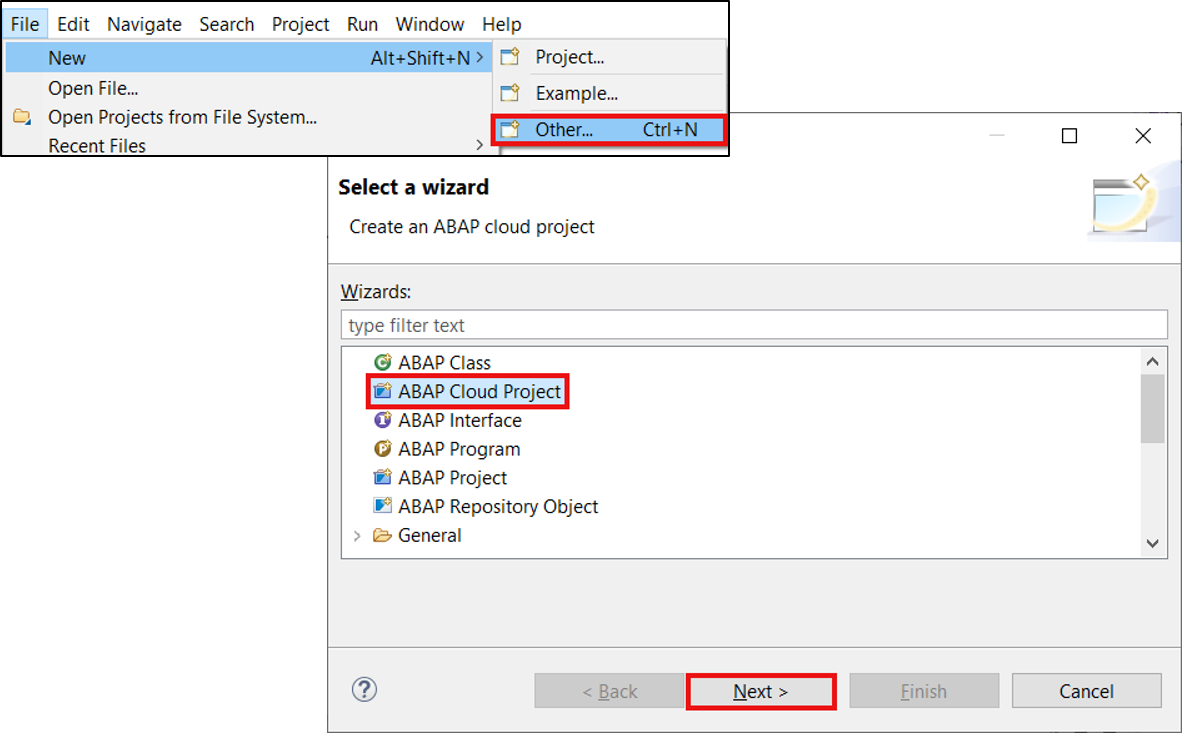
First, we need to install Eclipse IDE to perform the practical’s. We need to follow the following steps to configure Eclipse IDE on local system.

**To install the front-end component of ADT, proceed as follows:**

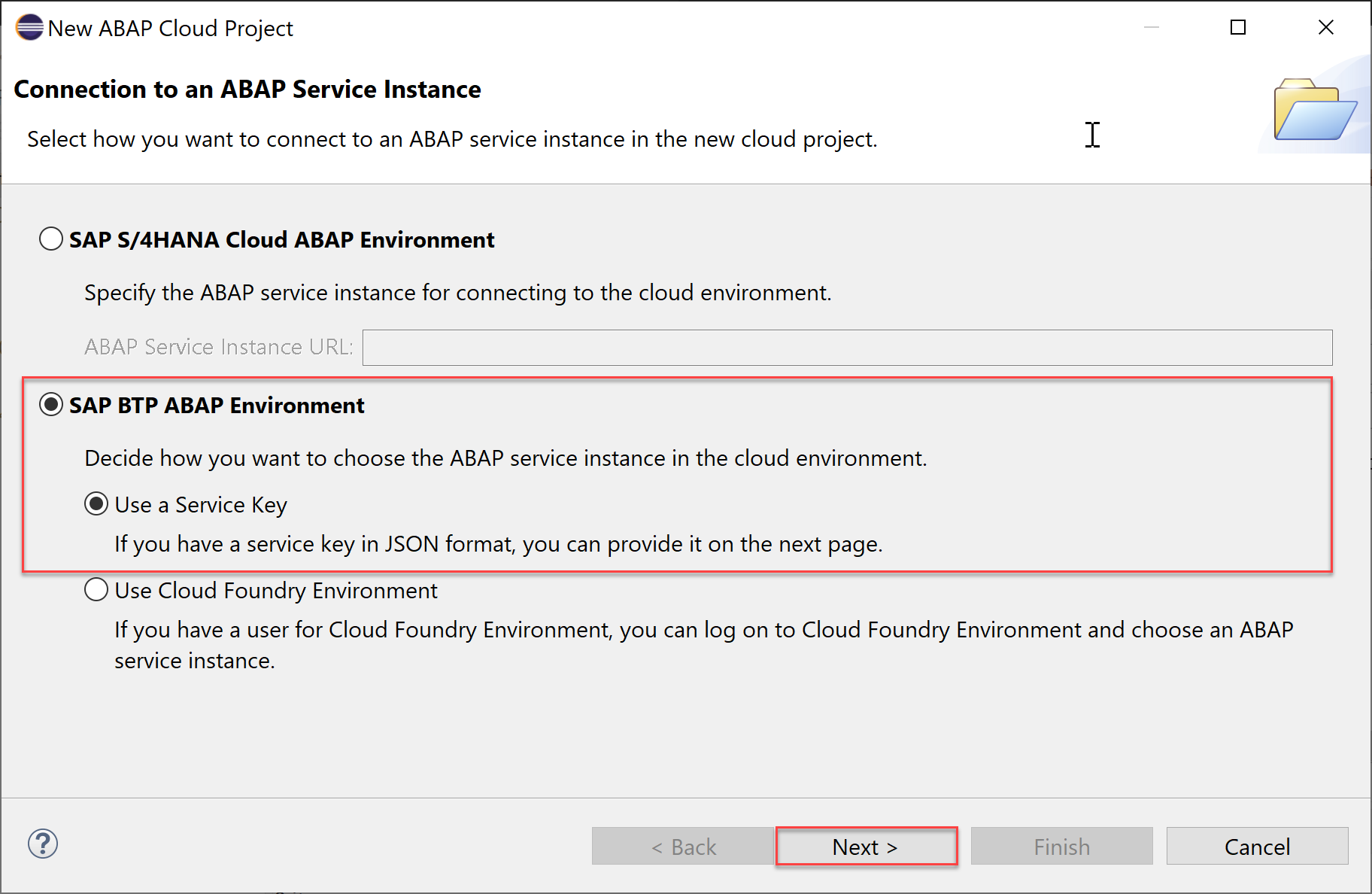
1. Get an installation of Eclipse 2022-09 (x86\_64) ([Eclipse IDE for Java Developers](file:///D:\Edunet\SAP%20Gujarat\Student%20Content\SAP%20BTP%20ABAP%20Environment\e.g.%20https:\www.eclipse.org\downloads\packages\release\2022-09\r\eclipse-ide-java-developers))
2. In Eclipse, choose in the menu bar Help > Install New Software...
3. Enter the URL https://tools.hana.ondemand.com/latest
4. Press Enter to display the available features.
5. Select ABAP Development Tools and choose next.
6. On the next wizard page, you get an overview of the features to be installed. Choose Next.
7. Confirm the license agreements and choose Finish to start the installation.

Step 3: Create ABAP cloud project

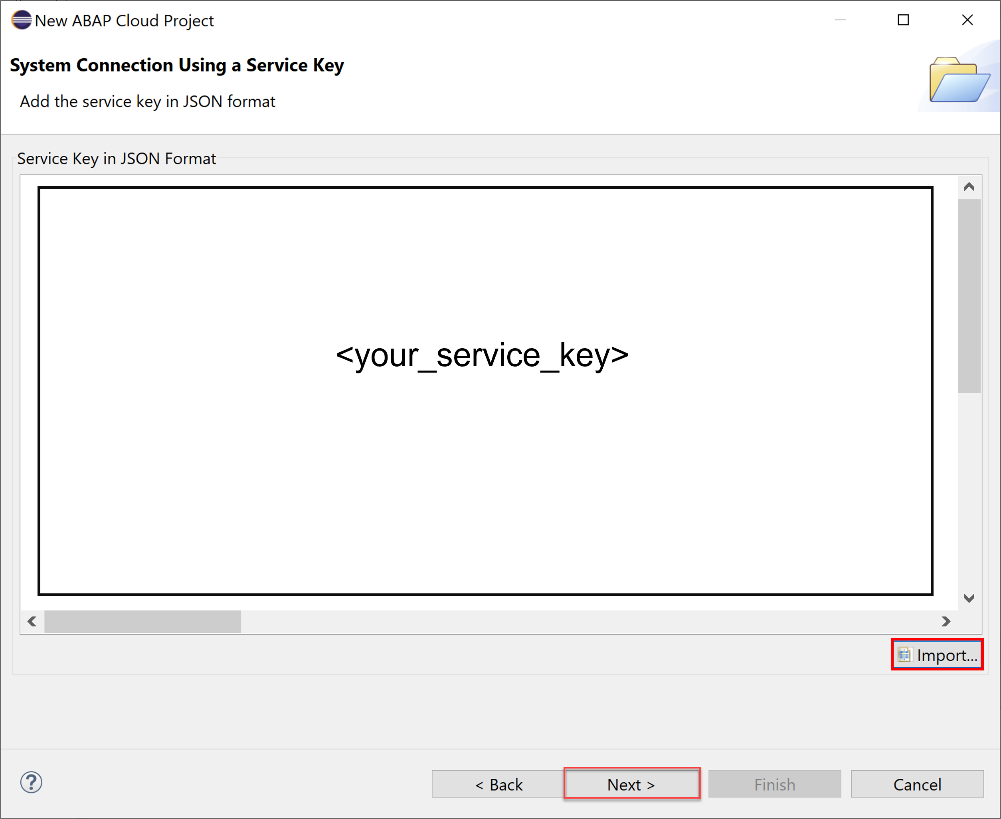
1. Select File > New > Other > ABAP Cloud Project and click Next >.



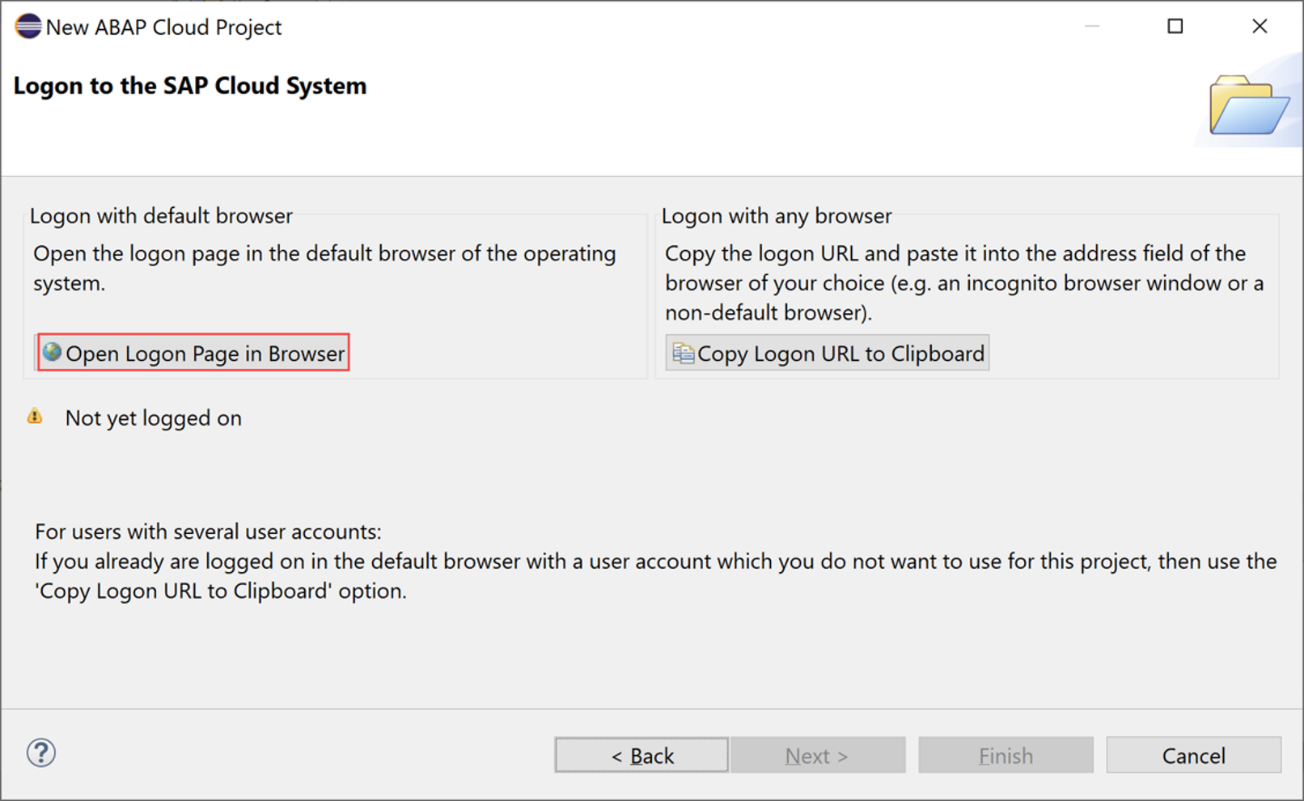
1. Select SAP BTP ABAP Environment > Use a Service Key and click Next >.



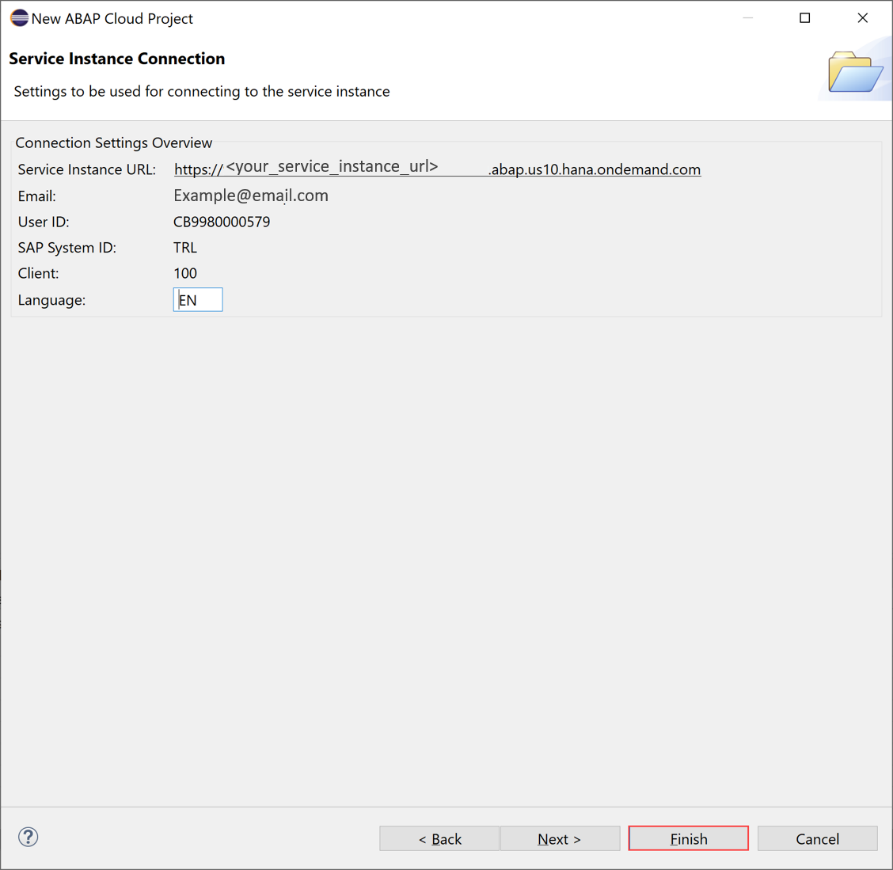
1. Import your service key and click Next >.



1. Click Open Logon Page in Browser.

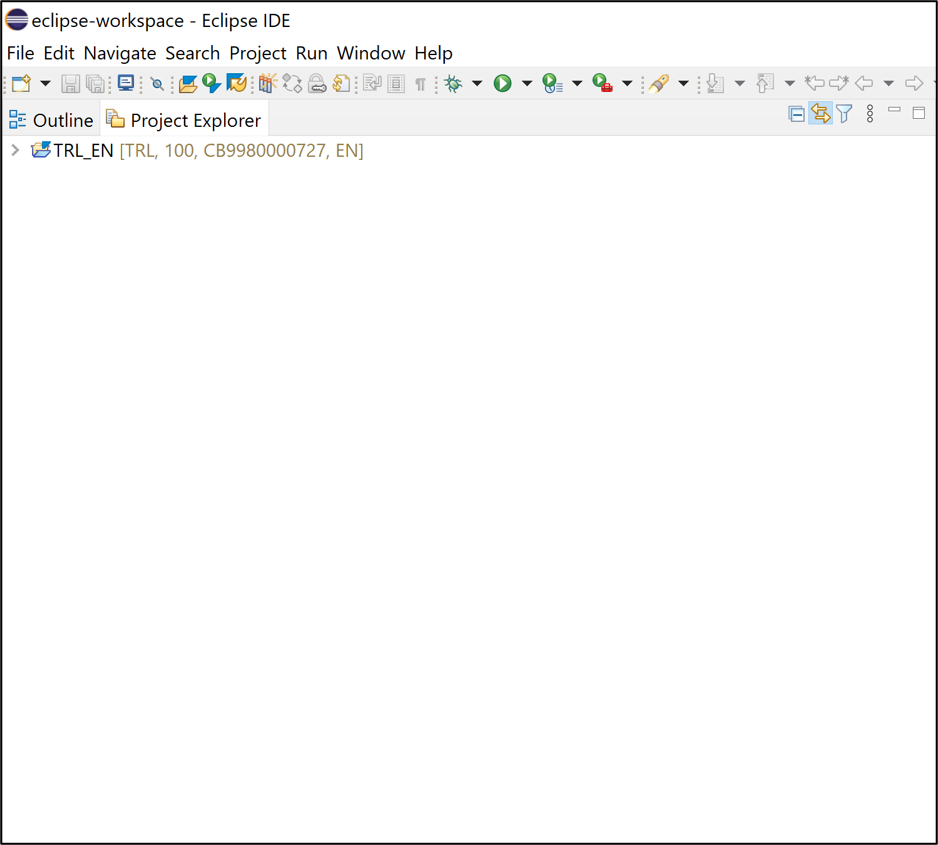


1. Now you’ve been authenticated automatically. Provide your credentials if requested. The credentials are the same you used to create your trial account on SAP BTP.



Click Finish.

1. Your trial system appears on the project explorer.



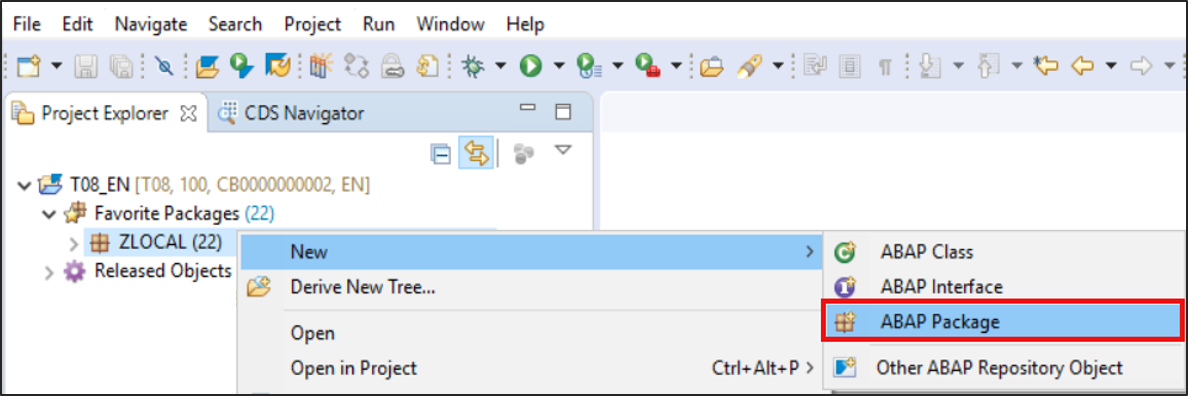
Lab 2 - Creating an ABAP Package

**In this Lab we will Learn about ABAP Package Creation**

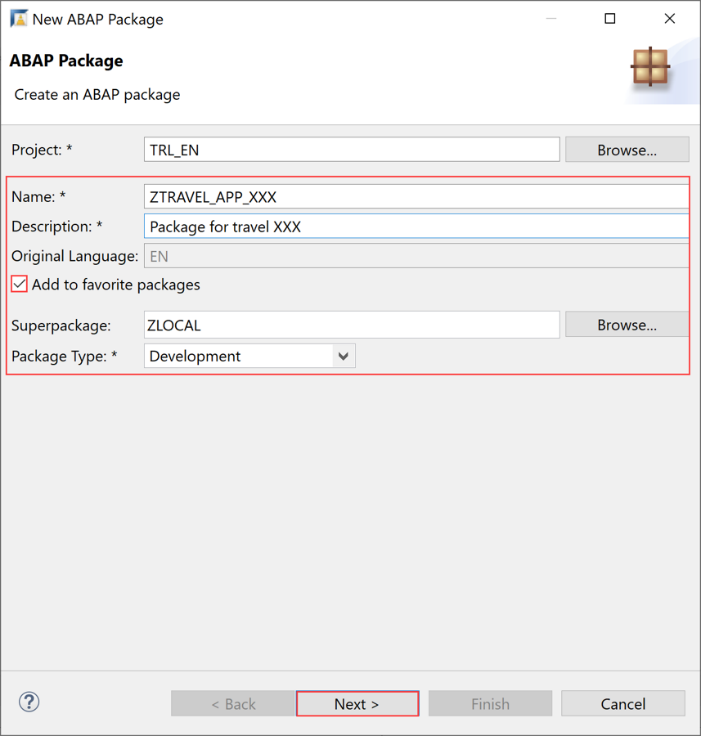
**Note: In this Lab, wherever XXX appears, use a number.**

* 1. Open ABAP Development Tools (ADT) and select your ABAP Cloud Project you created in Create a SAP BTP ABAP Environment Trial User.

Right-click on ZLOCAL and select New > ABAP Package.

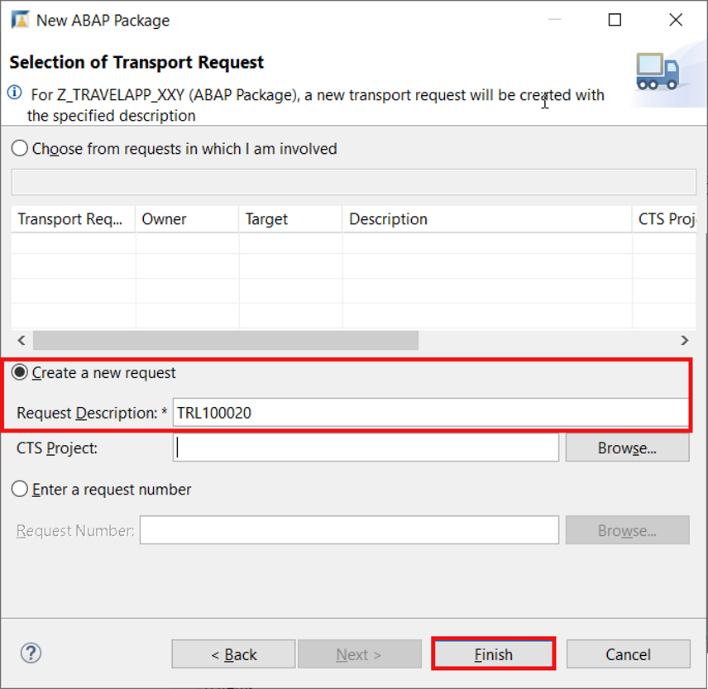


* 1. Create a new ABAP package:
  + Name: ZTRAVEL\_APP\_XXX
  + Description: Package for travel XXX
  + Superpackage: ZLOCAL
  + Check Add to favorite packages.



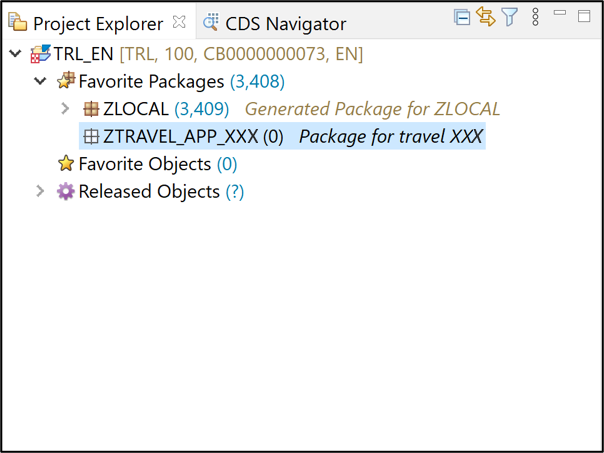
Click Next >.

* 1. Select Create new request and enter a request description.



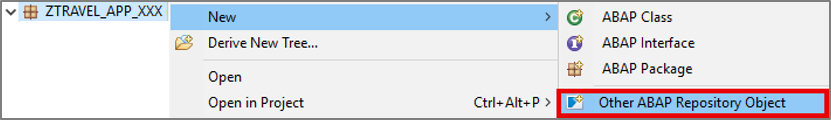
Click Finish.

* 1. Now your package is added to favorite objects.

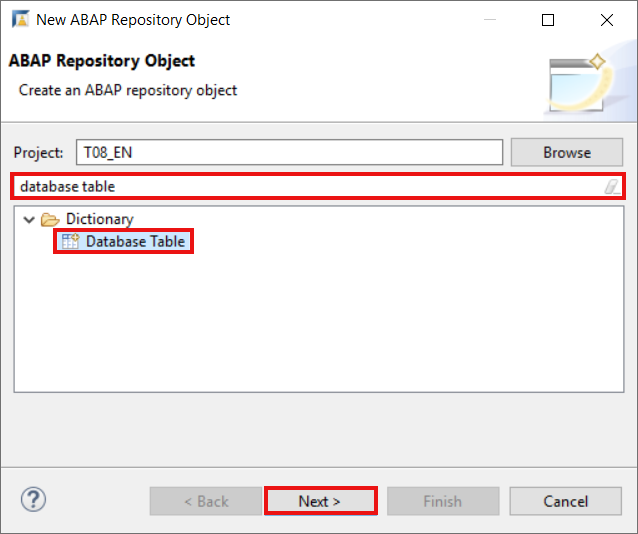


Lab 3 - Creating a Database Table

1. Right-click on your package ZTRAVEL\_APP\_XXX, select New > Other ABAP Repository Object.



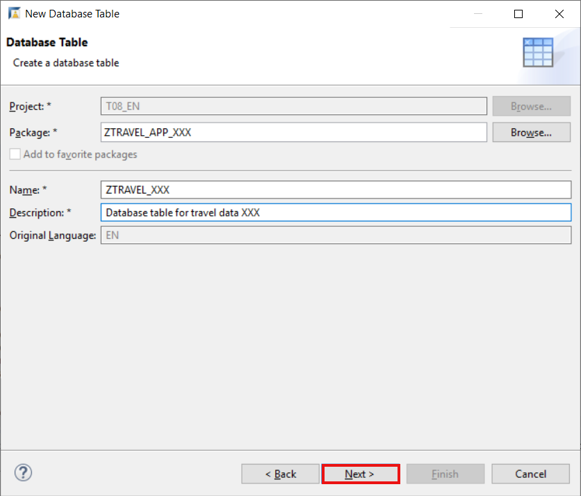
1. Search for database table, select it and click Next >.



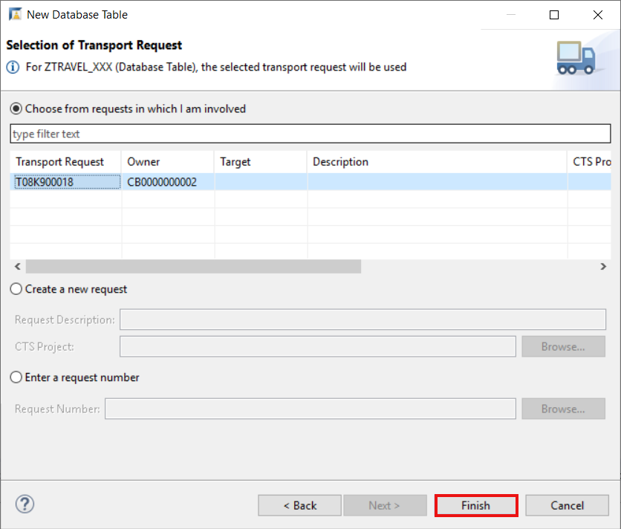
1. Create a new database table:

* Name: ZTRAVEL\_XXX
* Description: Database table for travel data XXX

Click Next >.



1. Click Finish to create your transport request.



1. Replace your code with following:

@EndUserText.label : 'Database table for travel data XXX'

@AbapCatalog.enhancementCategory : #NOT\_EXTENSIBLE

@AbapCatalog.tableCategory : #TRANSPARENT

@AbapCatalog.deliveryClass : #A

@AbapCatalog.dataMaintenance : #RESTRICTED

define table ztravel\_xxx {

key client : abap.clnt not null;

key mykey : sysuuid\_x16 not null;

travel\_id : /dmo/travel\_id;

agency\_id : /dmo/agency\_id;

customer\_id : /dmo/customer\_id;

begin\_date : /dmo/begin\_date;

end\_date : /dmo/end\_date;

@Semantics.amount.currencyCode : 'ztravel\_xxx.currency\_code'

booking\_fee : /dmo/booking\_fee;

@Semantics.amount.currencyCode : 'ztravel\_xxx.currency\_code'

total\_price : /dmo/total\_price;

currency\_code : /dmo/currency\_code;

description : /dmo/description;

overall\_status : /dmo/overall\_status;

created\_by : syuname;

created\_at : timestampl;

last\_changed\_by : syuname;

last\_changed\_at : timestampl;

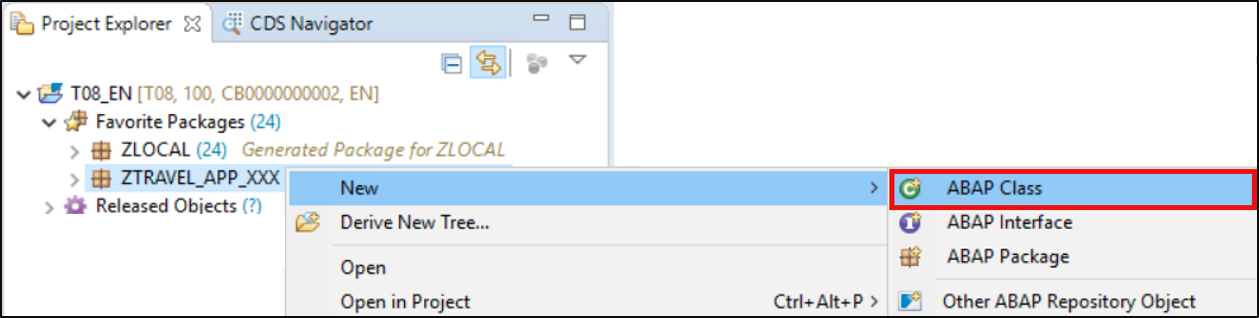
}

1. Save and Activate

* Save – Ctrl + S
* Activate – Ctrl+F3

Lab 4 - Create an ABAP Class

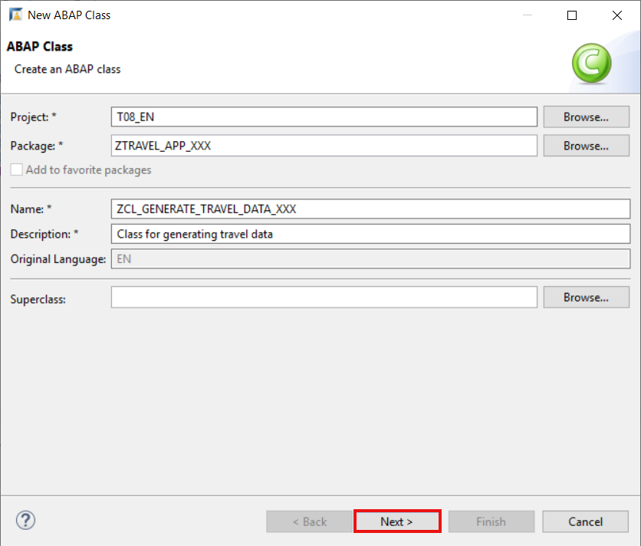
1. Right-click on your package ZTRAVEL\_APP\_XXX, select New > ABAP Class.



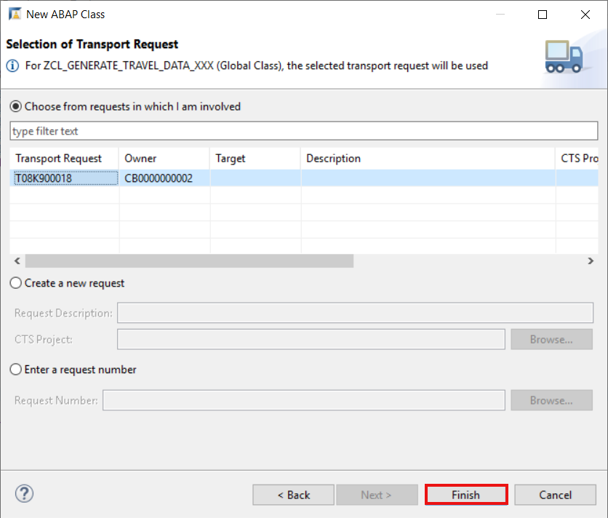
1. Create a new ABAP class:

* Name: ZCL\_GENERATE\_TRAVEL\_DATA\_XXX
* Description: Class for generating travel data

Click Next >.



1. Click Finish to create your transport request.



Lab 5 - WAP to print Hello World

**Create a New Class “zcl\_generate\_travel\_data\_yyy” and paste the following code.**

CLASS zcl\_generate\_travel\_data\_yyy DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES if\_oo\_adt\_classrun.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS zcl\_generate\_travel\_data\_yyy IMPLEMENTATION.

METHOD if\_oo\_adt\_classrun~main.

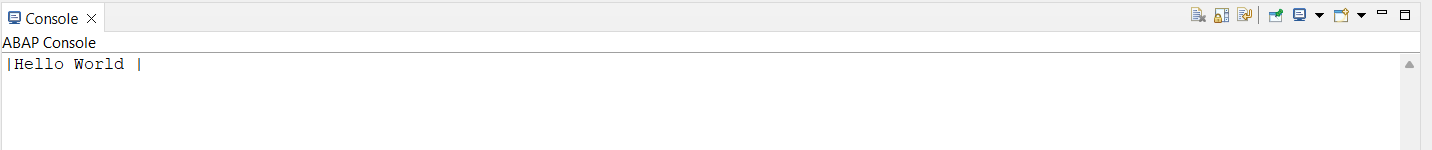
\* output the result as a console message.

out->write( '|Hello World |').

ENDMETHOD.

ENDCLASS.

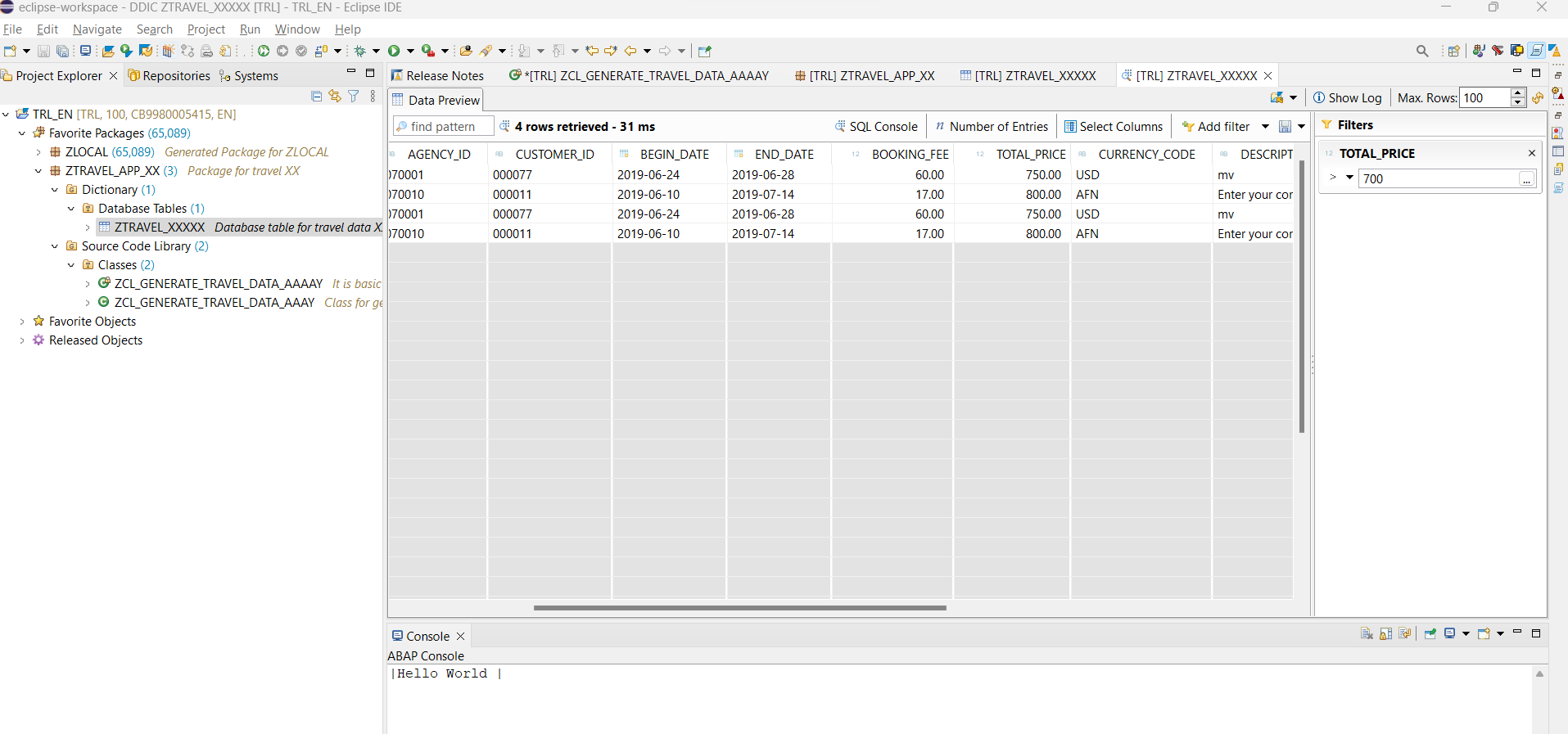
**Output:**



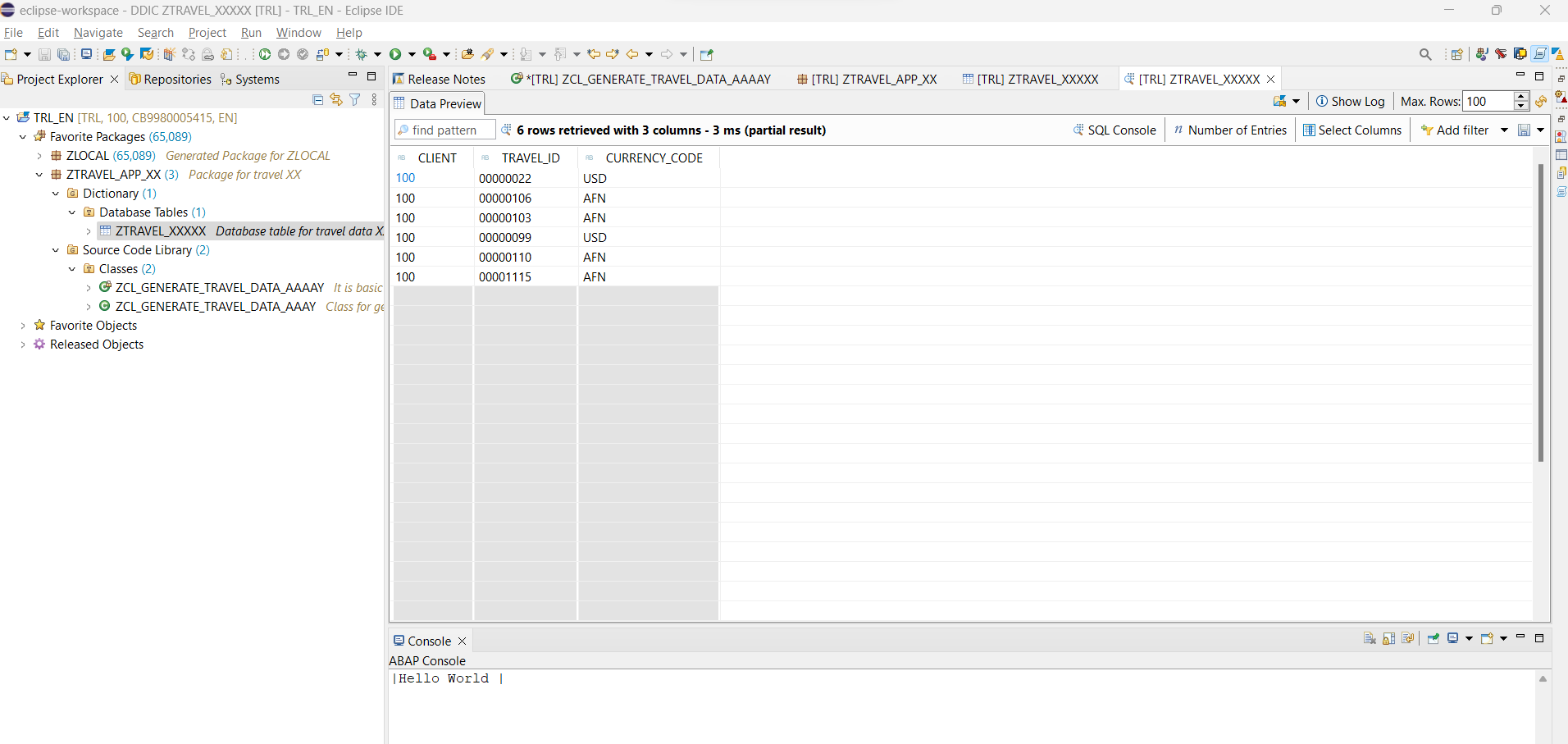
Lab 6 - Perform Query Operation on the Output Table

1. **How to extract the records having Total Price > 700**

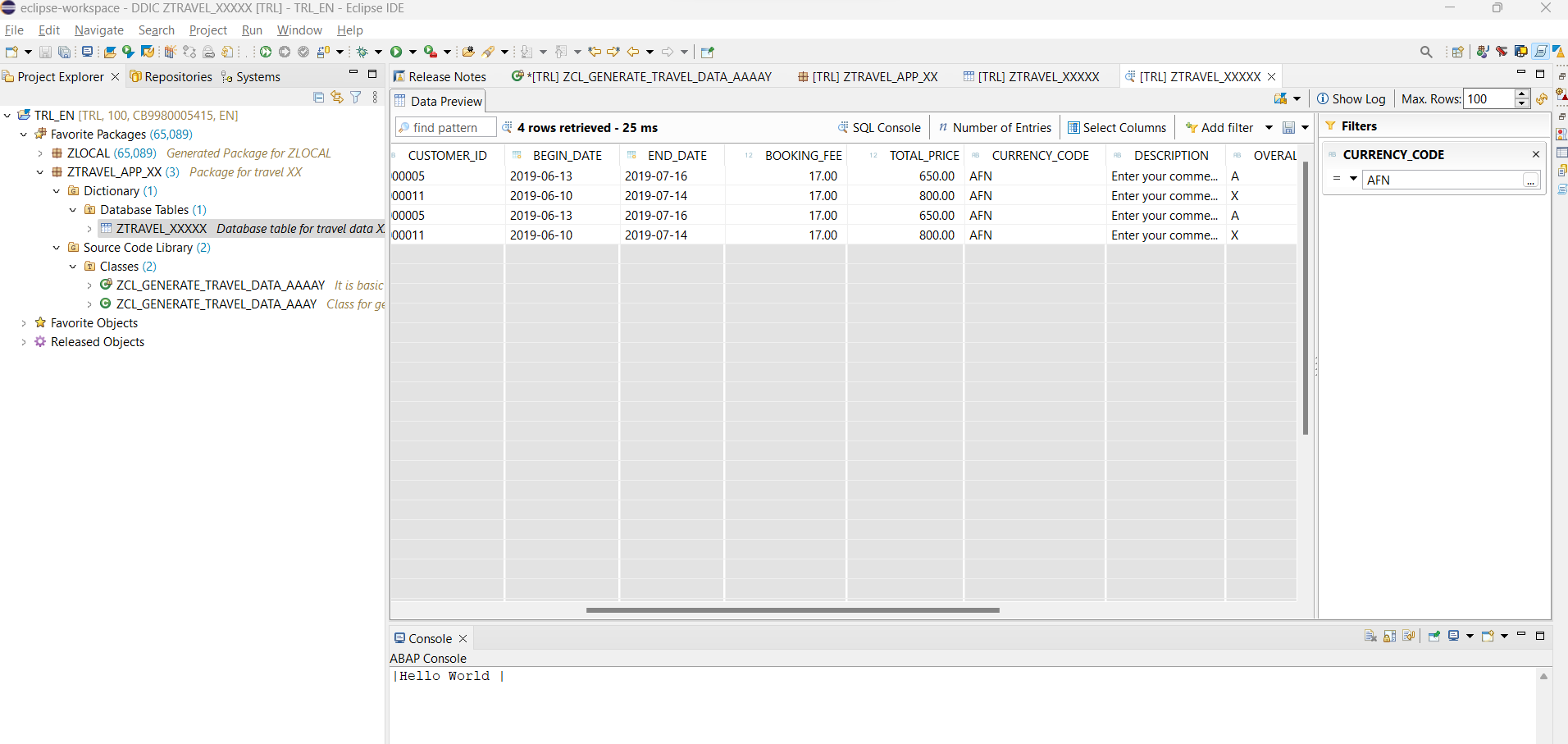
Click on “Add Filter”. Click on Total Price and write the condition



1. Click on Select Columns and Select the columns you want to display



1. Filter the Country Code = ‘AFN’ from the data



Lab 7 - Perform various ABAP data types of operation

Numeric Operations:

Addition:

DATA lv\_num1 TYPE i VALUE 10.

DATA lv\_num2 TYPE i VALUE 5.

DATA lv\_sum TYPE i.

lv\_sum = lv\_num1 + lv\_num2.

WRITE: 'Sum:', lv\_sum.

Subtraction:

DATA lv\_num1 TYPE i VALUE 10.

DATA lv\_num2 TYPE i VALUE 5.

DATA lv\_diff TYPE i.

lv\_diff = lv\_num1 - lv\_num2.

WRITE: 'Difference:', lv\_diff.

Multiplication:

DATA lv\_num1 TYPE i VALUE 10.

DATA lv\_num2 TYPE i VALUE 5.

DATA lv\_product TYPE i.

lv\_product = lv\_num1 \* lv\_num2.

WRITE: 'Product:', lv\_product.

Division:

DATA lv\_num1 TYPE f VALUE 10.0.

DATA lv\_num2 TYPE f VALUE 5.0.

DATA lv\_quotient TYPE f.

lv\_quotient = lv\_num1 / lv\_num2.

WRITE: 'Quotient:', lv\_quotient.

2-Character/String Operations:

Concatenation:

DATA lv\_str1 TYPE string VALUE 'Hello'.

DATA lv\_str2 TYPE string VALUE 'World'.

DATA lv\_result TYPE string.

CONCATENATE lv\_str1 lv\_str2 INTO lv\_result.

WRITE: 'Result:', lv\_result.

String Length:

DATA lv\_str TYPE string VALUE 'Hello World'.

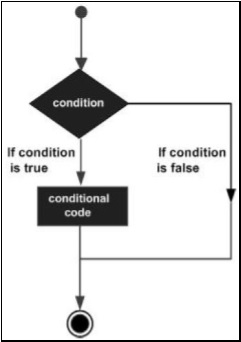
DATA lv\_length TYPE i.

lv\_length = STRLEN( lv\_str ).

WRITE: 'Length:', lv\_length.

ABAP programming language for decision-making statements.

the general form of a typical decision-making structure found in most of the programming languages −



|  |  |
| --- | --- |
| **S.No.** | **Statement & Description** |
| 1 | [IF Statement](https://www.tutorialspoint.com/sap_abap/sap_abap_if_statement.htm)  An IF statement consists of a logical expression followed by one or more statements. |
| 2 | [IF.. Else Statement](https://www.tutorialspoint.com/sap_abap/sap_abap_if_else_statement.htm)  An IF statement can be followed by an optional ELSE statement that executes when the expression is false. |
| 3 | [Nested IF Statement](https://www.tutorialspoint.com/sap_abap/sap_abap_nested_if_statement.htm)  You may use one IF or ELSEIF statement inside another IF or ELSEIF statement. |
| 4 | [CASE Control Statement](https://www.tutorialspoint.com/sap_abap/sap_abap_case_control_statement.htm)  CASE statement is used when we need to compare two or more fields or variables. |

Example

DATA lv\_age TYPE i.

lv\_age = 25.

IF lv\_age >= 18.

WRITE: 'You are an adult.'.

ELSE.

WRITE: 'You are not an adult.'.

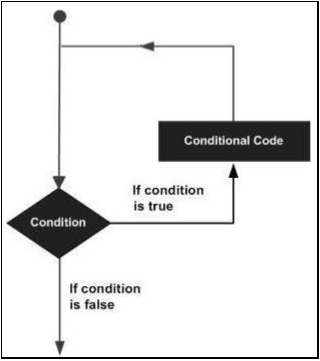
ENDIF.

In this example, we have a variable lv\_age assigned a value of 25. The IF statement checks whether lv\_age is greater than or equal to 18. If the condition is true, it executes the statements within the IF block and displays "You are an adult." Otherwise, it executes the statements within the ELSE block and displays "You are not an adult."

The output will be: ---- You are an adult.

Lab 8 - Loops in ABAP, For loop, While Loop

Programming languages provide various control structures that allow for more complicated execution paths. A **loop statement** allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages.



Code For Loop Example:

DATA: lv\_counter TYPE i.

lv\_counter = 1.

DO 5 TIMES.

WRITE: 'Counter:', lv\_counter.

lv\_counter = lv\_counter + 1.

ENDDO.

Explanation: In this example, we have a variable lv\_counter initialized with a value of 1. The DO loop is executed 5 times. Within the loop, the current value of lv\_counter is displayed using the WRITE statement. After each iteration, the value of lv\_counter is incremented by 1 using the assignment statement lv\_counter = lv\_counter + 1.

The output will be:

Counter: 1

Counter: 2

Counter: 3

Counter: 4

Counter: 5

While Loop Example:

DATA: lv\_counter TYPE i.

lv\_counter = 0.

WHILE lv\_counter < 5.

WRITE: 'Counter:', lv\_counter.

lv\_counter = lv\_counter + 1.

ENDWHILE.

**Explanation:** In this example, the WHILE loop is executed as long as lv\_counter is less than 5. The loop starts with lv\_counter initialized to 0, and within the loop, the current value of lv\_counter is displayed using the WRITE statement. After each iteration, the value of lv\_counter is incremented by 1 using the assignment statement lv\_counter = lv\_counter + 1.

The output will be:

Counter: 0

Counter: 1

Counter: 2

Counter: 3

Counter: 4

Lab 9 - **Insert Data Table Entries and print them on the console**

**Open the class first created “zcl\_generate\_travel\_data\_xxx” and paste the following code.**

CLASS zcl\_generate\_travel\_data\_xxx DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES if\_oo\_adt\_classrun.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS zcl\_generate\_travel\_data\_xxx IMPLEMENTATION.

METHOD if\_oo\_adt\_classrun~main.

DATA itab TYPE TABLE OF ztravel\_xxx.

\* fill internal travel table (itab)

itab = VALUE #(

( mykey = '02D5290E594C1EDA93815057FD946624' travel\_id = '00000022' agency\_id = '070001' customer\_id = '000077' begin\_date = '20190624' end\_date = '20190628' booking\_fee = '60.00' total\_price = '750.00' currency\_code = 'USD'

description = 'mv' overall\_status = 'A' created\_by = 'MUSTERMANN' created\_at = '20190612133945.5960060' last\_changed\_by = 'MUSTERFRAU' last\_changed\_at = '20190702105400.3647680' )

( mykey = '02D5290E594C1EDA93815C50CD7AE62A' travel\_id = '00000106' agency\_id = '070005' customer\_id = '000005' begin\_date = '20190613' end\_date = '20190716' booking\_fee = '17.00' total\_price = '650.00' currency\_code = 'AFN'

description = 'Enter your comments here' overall\_status = 'A' created\_by = 'MUSTERMANN' created\_at = '20190613111129.2391370' last\_changed\_by = 'MUSTERMANN' last\_changed\_at = '20190711140753.1472620' )

( mykey = '02D5290E594C1EDA93858EED2DA2EB0B' travel\_id = '00000103' agency\_id = '070010' customer\_id = '000011' begin\_date = '20190610' end\_date = '20190714' booking\_fee = '17.00' total\_price = '800.00' currency\_code = 'AFN'

description = 'Enter your comments here' overall\_status = 'X' created\_by = 'MUSTERFRAU' created\_at = '20190613105654.4296640' last\_changed\_by = 'MUSTERFRAU' last\_changed\_at = '20190613111041.2251330' )

).

\* delete existing entries in the database table

DELETE FROM ztravel\_xxx.

\* insert the new table entries

INSERT ztravel\_xxx FROM TABLE @itab.

\* output the result as a console message

out->write( |{ sy-dbcnt } travel entries inserted successfully!| ).

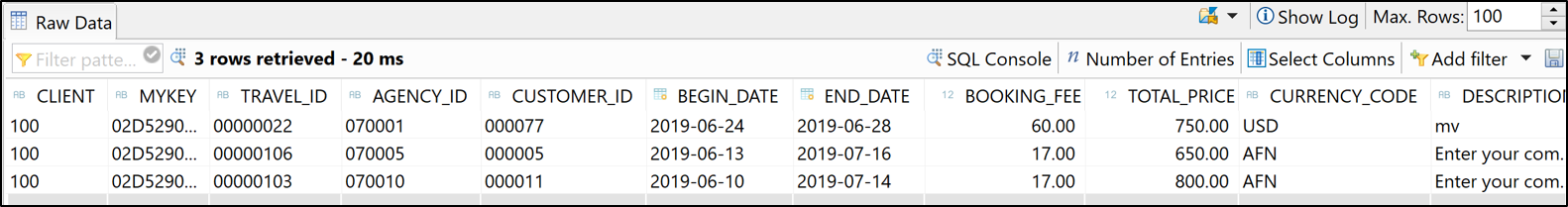
ENDMETHOD.

ENDCLASS.

1. Save, activate and click F9 to run your ABAP class.

* Save – Ctrl + S
* Activate – Ctrl + F3
* Run – F9

Check your result. Therefore, open your database table ZTRAVEL\_XXX and press F8 to see your data. Now the dictionary tables are filled with data.



Reference

1. <https://www.sap.com/india/about/company/what-is-sap.html>
2. <https://www.investopedia.com/terms/e/erp.asp>
3. <https://www.g2.com/categories/erp-systems>
4. <https://learning.sap.com/learning-journey/get-started-with-abap-programming-on-sap-btp/understanding-the-basic-features-of-abap_c0e5346f-a136-4b9f-a167-903>
5. <https://www.sap.com/documents/2018/08/7a62516c-157d-0010-87a3-c30de2ffd8ff.html>
6. 29.<https://open.sap.com/courses/mm4h2>
7. 30.<https://www.udemy.com/topic/sap-mm/>
8. 31.<https://blogs.sap.com/2008/05/09/sap-project-system-a-ready-reference-part-1>
9. 32.<https://training.sap.com/content/sap-training-hana>
10. 33.<https://learning.sap-press.com/abap>
11. 34.<https://data-flair.training/blogs/sap-hana-vs-sap-s-4-hana/>
12. 35.<https://axxis-consulting.com/what-is-sap-c4hana/>