

## Flexible Search Queries [FSQ] =

Let's say we have DB →

**Q =** What are the different activities we perform?

1) **DDL** =

- a. Create table
- b. Alter table
- c. Create Cols
- d. Alter Cols
- e. =====

**Note** = In "SAP Comm" how / where we do this? = **\*-items.xml**

2) **DML**

a. Inserting record

**Q1** = In "SAP Comm" how we do this? = **ImpEx**

b. Updating record

**Q2** = In "SAP Comm" how we do this? = **ImpEx**

c. Deleting record

**Q3** = In "SAP Comm" how we do this? = **ImpEx**

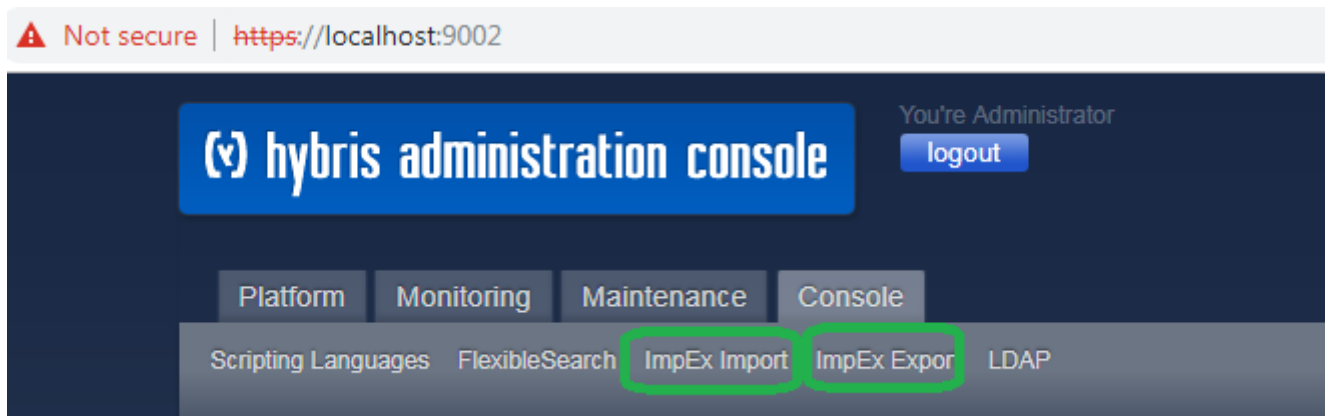
**Note** = We are writing the Impex – Means – U might be Inserting / Updating / Deleting the Record.

ImpEx → Import ImpEx =

We are getting the records into "SAP Comm".

ImpEx → Export ImpEx =

We are exporting the records from "SAP Comm"



**d. Getting records (Oracle – Select)**

**Q4** = In “SAP Comm” how we do this? =

Flexible Search Queries (**FSQ**)

We are writing the FSQ – Means – You are getting the Records.

==> **Oracle – Select** === “**SAP Comm**” – **FSQ**

**Oracle – Select** command Syntax: -

Select \* (or) EmpId, EmpName from **Emp** (TableName)

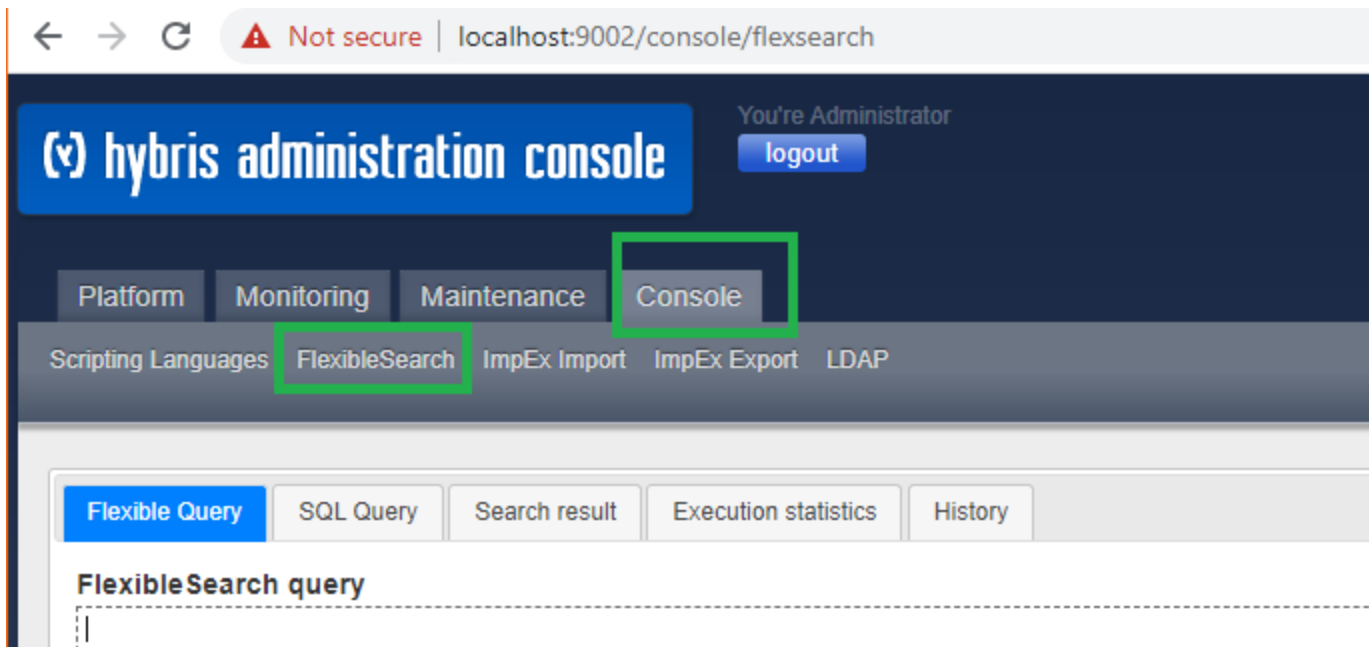
[Where ==] [Order by ==] [Group by ==]

→ All these are applicable for “SAP Comm” also.

**Q** = How to execute the FSQ?

Sol1 = **hAC**

Sol2 = **Programatically**

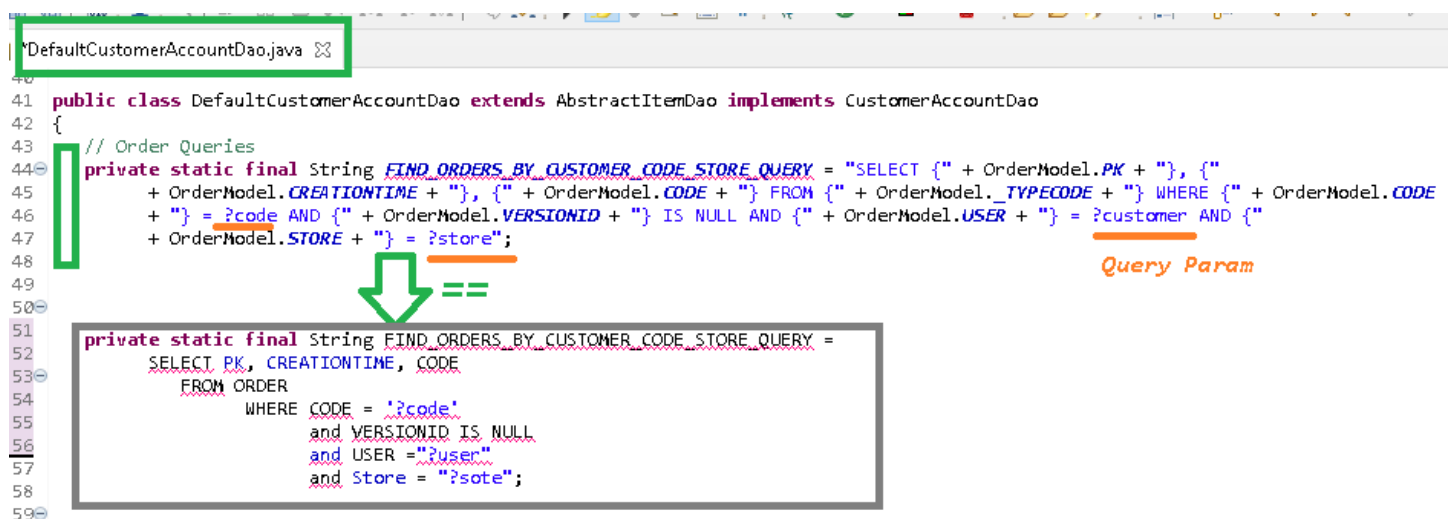


**Note** = In “SAP Comm” we write the FSQ. Internally this will be executed in 2 phases: -

- 1) **Phase 1** = FSQ will be converted into equivalent SQL query
- 2) **Phase 2** = This converted SQL query will be executed.

**Programmatically =**

**Step 1 = Write the Query**



## Step 2 = Set the Params

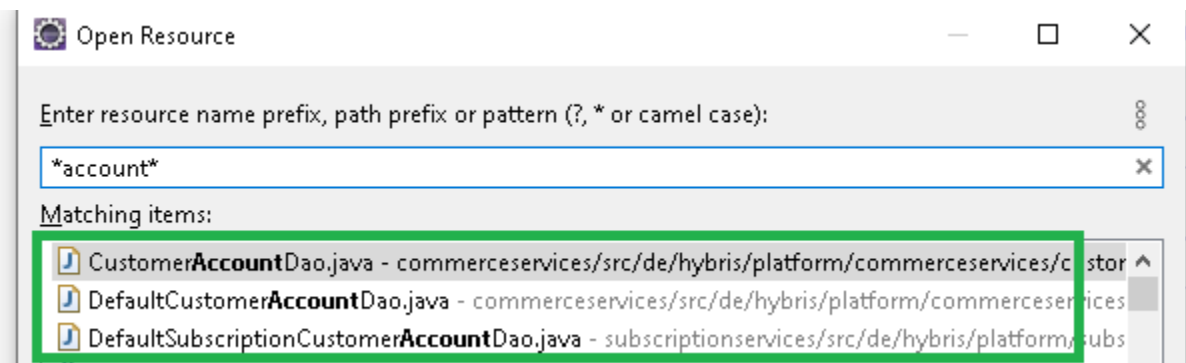
```
DefaultCustomerAccountDao.java
123 @Override
124 public OrderModel findOrderByCustomerAndCodeAndStore(final CustomerModel customerModel, final String code,
125 final BaseStoreModel store)
126 {
127     validateParameterNotNull(customerModel, "Customer must not be null");
128     validateParameterNotNull(code, "Code must not be null");
129     validateParameterNotNull(store, "Store must not be null");
130     final Map<String, Object> queryParams = new HashMap<String, Object>();
131     queryParams.put("customer", customerModel);
132     queryParams.put("code", code);
133     queryParams.put("store", store);
134     final OrderModel result = getFlexibleSearchService().searchUnique(
135         new FlexibleSearchQuery(FIND_ORDERS_BY_CUSTOMER_CODE_STORE_QUERY, queryParams));
136     return result;
137 }
138
```

## Step 3 = Execute FSQ

```
DefaultCustomerAccountDao.java
123 @Override
124 public OrderModel findOrderByCustomerAndCodeAndStore(final CustomerModel customerModel, final String code,
125 final BaseStoreModel store)
126 {
127     validateParameterNotNull(customerModel, "Customer must not be null");
128     validateParameterNotNull(code, "Code must not be null");
129     validateParameterNotNull(store, "Store must not be null");
130     final Map<String, Object> queryParams = new HashMap<String, Object>();
131     queryParams.put("customer", customerModel);
132     queryParams.put("code", code);
133     queryParams.put("store", store);
134
135     final OrderModel result = getFlexibleSearchService().searchUnique(
136         new FlexibleSearchQuery(FIND_ORDERS_BY_CUSTOMER_CODE_STORE_QUERY, queryParams));
137
138     return result;
139 }
140
```

Query                      Params

**Note** = Generally, we write the FSQ in “\***DAO.java**” file.



The screenshot shows the SAP IDE interface. On the left, the 'Project Explorer' displays a list of files, with 'core-items.xml' selected and marked with a green checkmark. The main editor shows the XML content of 'core-items.xml'. A green arrow points to the line `<!-- product` with the annotation 'Used for "SAP Comm"'. Another green arrow points to the line `<itemtype code="Product"` with the annotation 'Used for DB'. The XML snippet is as follows:

```

3379
3380
3381
3382
3383 <!-- product
3384 <itemtype code="Product"
3385     extends="GenericItem"
3386     jaloclass="de.hybris.platform.jalo.product.Product"
3387     autocreate="true"
3388     generate="true">
3389     <deployment table="Products" typecode="1" propertytable="ProductProps"/>
3390     <attributes>

```

On the right, a table view shows the data for the 'Products' table. The table has columns: HJMPTS, CREATEDTS, MODIFIEDTS, and TYPEI. The data is as follows:

HJMPTS	CREATEDTS	MODIFIEDTS	TYPEI
17	2018-12-05 15:35:08.195	2018-12-05 15:36:11.276	87960988
17	2018-12-05 15:35:08.198	2018-12-05 15:36:11.278	87960988
26	2018-12-05 15:35:08.202	2018-12-05 15:36:39.14	87960988
25	2018-12-05 15:35:08.205	2018-12-05 15:36:11.282	87960988
26	2018-12-05 15:35:08.208	2018-12-05 15:36:39.132	87960988
23	2018-12-05 15:35:08.212	2018-12-05 15:39:09.537	87960988
17	2018-12-05 15:35:08.215	2018-12-05 15:36:11.289	87960988
11	2018-12-05 15:35:08.219	2018-12-05 15:36:11.291	87960988
15	2018-12-05 15:35:08.222	2018-12-05 15:36:11.293	87960988
11	2018-12-05 15:35:08.226	2018-12-05 15:36:11.295	87960988

**Product (Table) =**

ProductID	ProductName	ProductPrice	=== Cols
=====	=====	=====	
P101	ABC	10\$	== Record 1
P102	XYZ	20\$	== Record 2

**Q1 =** Get all the product table records?

**Oracle** = select \* from product;

**SAP Comm** = select {pk} from {product}

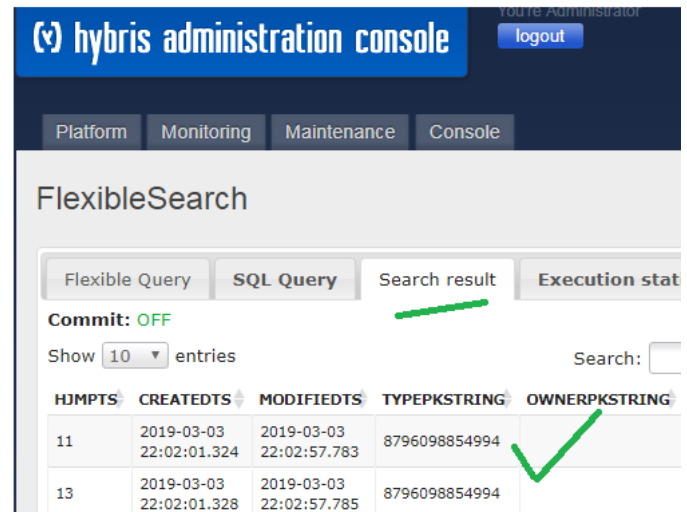
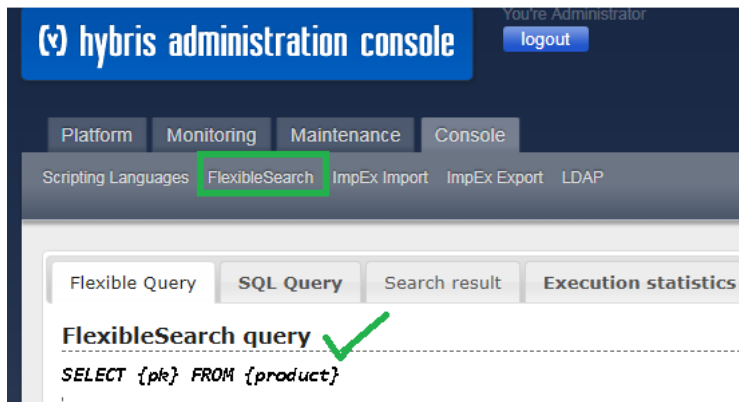
{product} = itemtype name.

Itemtype names & Cols name – We need to put inside { }

**Q2** = Get only “Product Code & ManufacturerName” for all the Products?

**Oracle** = select code, manufacturername from products;

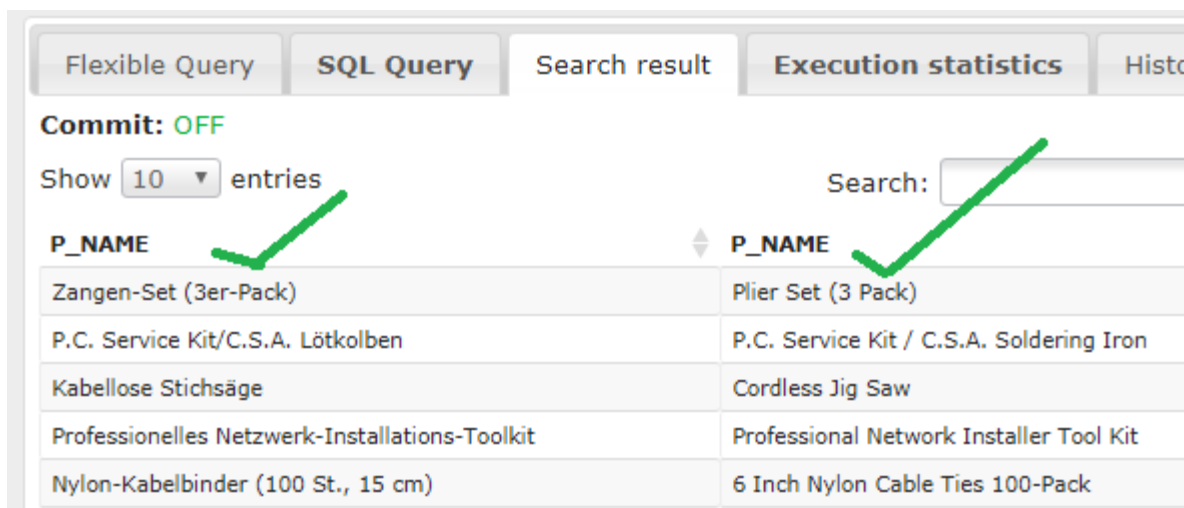
**SAP Comm** = select {code}, {manufacturername} from {product}



**Q3** = Product name is localized.

How to get localized names for all the Products?

select {name[de]}, {name[en]} from {product}



**Q4** = How to give the Alias Names?

**Oracle** = select p.code, p.manufacturername from products **as p**;

**SAP Comm** =

select {p.code}, {p.manufacturername} from {product **as p**}

**Note** = FSQ will be executed in 2 phases: -

1) We write FSQ & it will be converted into equivalent native query

The screenshot shows the FlexibleSearch console at <https://localhost:9002/console/flexsearch>. The 'Flexible Query' tab is active, displaying the query: `select {name[de]}, {name[en]} from {product}`. The 'Direct SQL query' tab is also visible, showing the converted SQL: `SELECT lp_t0.p_name , lp_t0.l1.p_name FROM productslp lp_t0 JOIN productslp lp_t0.l1 ON lp_t0.ITEMPK = lp_t0.l1.ITEMPK AND lp_t0.l1.LANGPK =? WHERE ((lp_t0.LANGPK =? )) AND (lp_t0.ITEMTYPEPK IN (?,?,?,?,?,?,?))`. Red boxes highlight the query input and the resulting SQL.

2) This converted query will be executed & get the results.

The screenshot shows the FlexibleSearch console with the 'Search result' tab active. The 'Commit' button is highlighted in green. The search results are displayed in a table with two columns: 'P\_NAME' and 'P\_NAME'. The results are:

P_NAME	P_NAME
Zangen-Set (3er-Pack)	Plier Set (3 Pack)
P.C. Service Kit/C.S.A. LötKolben	P.C. Service Kit / C.S.A. Soldering Iron
Kabellose Stichsäge	Cordless Jig Saw

**Q5** = Count How many Products are there?

**Oracle** = select **count(\*)** from Products

**SAP Comm** = select count({pk}) from {product}

**Note** = We can apply all math's functions (count, min, max, sum, avg ....).

**Q6** = Get all the Products which contains Product **code** = **011**?

**Oracle** = select \* from products where code like '%011%'

**SAP Comm** =

select {code} from {product} **where {code} like '%011%'**

select {code} from {product} where {code} **not like '%011%'**

select {code} from {product} where {code} like '%011%' **or** {code} like '%11%'

select {name} from {product} **where {name[de]} is not null**

**Note** = We can apply all the condition operators (**like ... not like ... and ... or ... is null ... is not null ...**.)

**Q7** = Display all the product codes in ASC order from Product table?

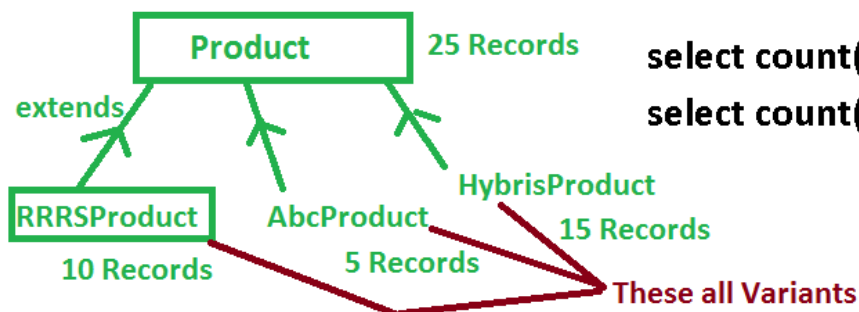
**Oracle** = select {code} from {product} order by {code}

**SAP Comm** = select {code} from {product} **order by {code} desc**

**Note** = Performance improvement query.

**select count({pk}) from {product}** = Showing 7,158 entries

**select count({pk}) from {product!}** = Showing 1,431 entries

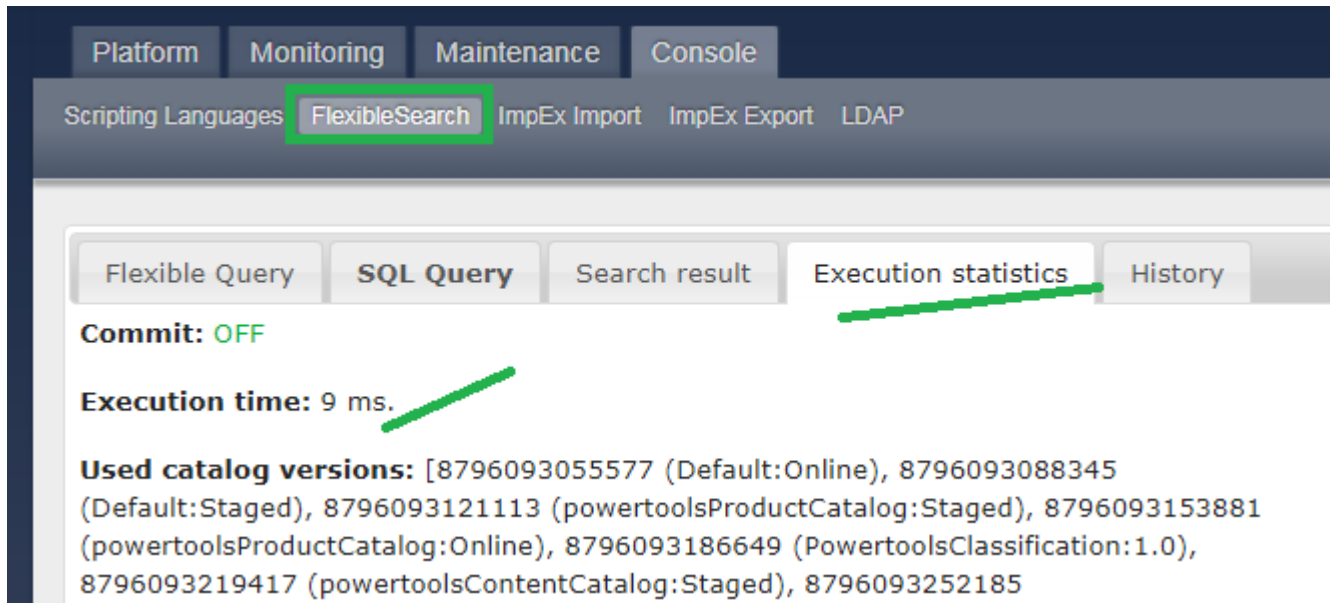


**select count({pk}) from {product}** = 55 Records

**select count({pk}) from {product!}** = 25 Records



**Q8** = Where can we see execution time of the query?



**Note** = We can also apply the joins.

**Q** = What are the different types of joins?

- 1) Left outer join
- 2) Right outer join
- 3) Inner join
- 4) Full join

**Q9** = Get all user groups by parent unit?

```
SELECT * FROM {B2BUserGroup as ug JOIN B2BUnit as u ON {ug:unit} = {u:pk} }a
```

**Q10** = Get all product codes from product and category table using left join?



**Q** = Is this a valid Flexible Search Query? =

SELECT {p.pk} FROM {Principal} AS p WHERE {p.uid} = 'admin'

- No. Bcoz, Principal is an abstract type
- No. Bcoz, SQL aliases definition needs to be inside brackets
- No - for another reason
- Yes, this is a valid Flexible Search Query.

Combined SELECT Statements with UNION Operator =

```
SELECT x.PK FROM ( {{SELECT {PK} as PK FROM {Chapter} WHERE {Chapter.PUBLICATION} LIKE 6587084167216798848 }}
UNION ALL
{{ SELECT {PK} as PK FROM {Page} WHERE {Page.PUBLICATION} LIKE 6587084167216798848 }})
```

Query = Get top 10 orders (the order that contains the max products)in specific time frame

```

select {order.code} as order_id, count({orderentry.product}) as products_sold
from {order}, {orderentry}
where {order.pk} = {orderentry.order}
and {orderentry.modifiedtime} >= to_date('2018/06/12','YYYY/MM/DD')
group by {order.code}
order by 2 desc
fetch first 10 rows only

```

output of this query

Flexible Query	SQL Query	Search result	Execution statistics	History
Commit: OFF				
Show 10 entries		Search: <input type="text"/>		
ORDER_ID	PRODUCTS_SOLD			
acceptanceTestOrder12	5			
acceptanceTestOrder1	4			
acceptanceTestOrder11	3			

Query = get the top 10 products which are sold in specific time frame

```

select {orderentry.product} as product_code, sum({orderentry.quantity}) as products_sold
from {orderentry}
where {orderentry.modifiedtime} >= to_date('2018/06/12','YYYY/MM/DD') and
{orderentry.modifiedtime} <= to_date('2018/06/14','YYYY/MM/DD')
group by {orderentry.product}
order by products_sold desc
fetch first 10 rows only

```

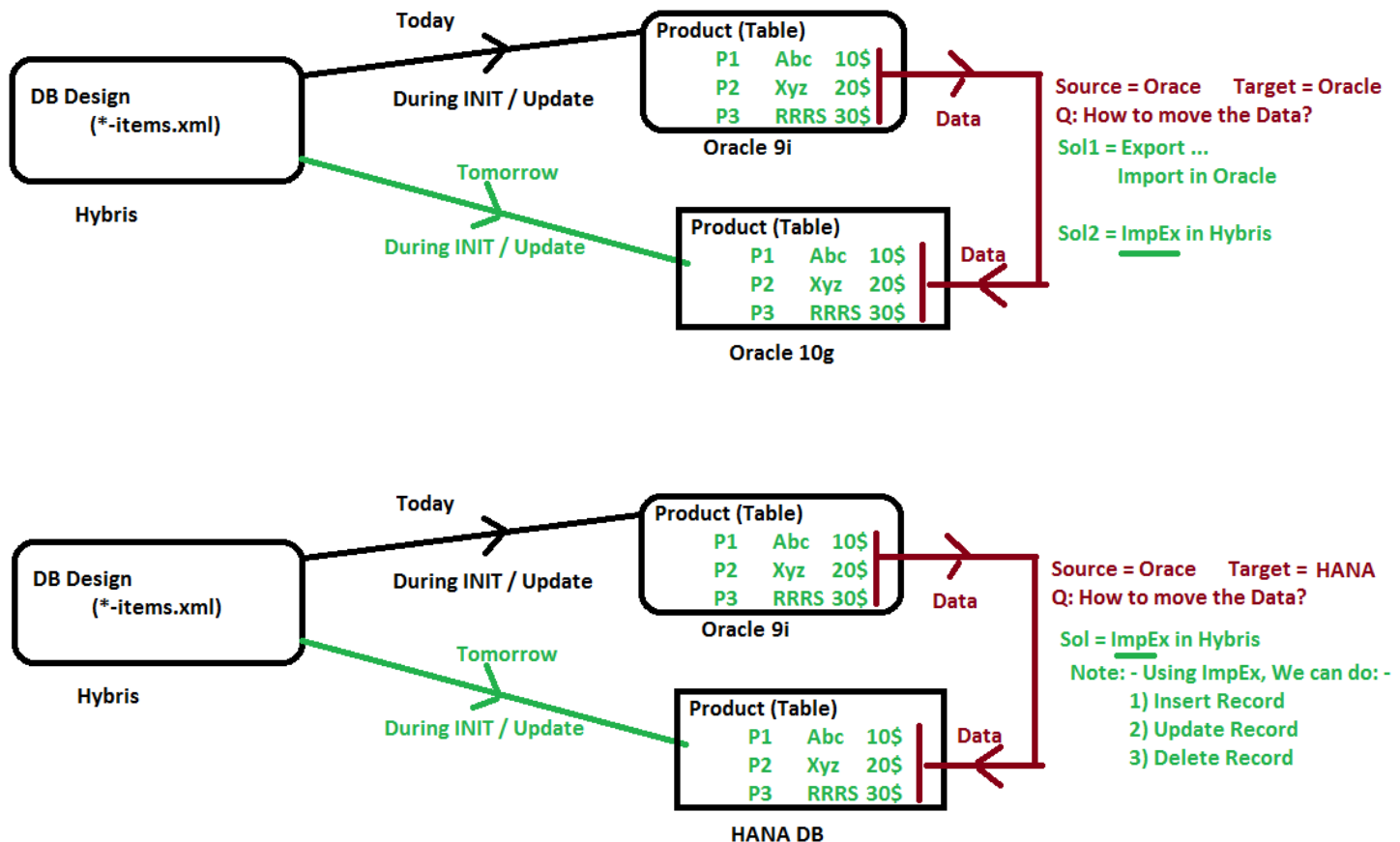
output of this query

Flexible Query	SQL Query	Search result	Execution statistics	History
Commit: OFF				
Show 10 entries		Search: <input type="text"/>		
PRODUCT_CODE	PRODUCTS_SOLD			
8796196175873	53.00000000			
8796195127297	36.00000000			
8796193914881	33.00000000			
8796194439169	1.00000000			
8796192571393	1.00000000			

## ImpEx

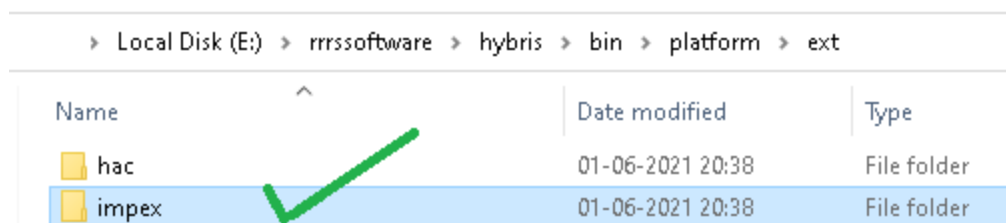
You are writing ImpEx means: -

- 1) You might be inserting / updating / deleting records = **Import ImpEx**
- 2) You might be taking out the “SAP Comm” records = **Export ImpEx**



**ImpEx** = This is text-based Import / Export.

“SAP Comm” provided Exts called “**ImpEx**”.



**Let's say** – We have a table called “**Emp**” in Oracle DB.

<b>EmpId</b>	<b>EmpName</b>	<b>EmpSal</b>
101	RRRS	10
102	Chenna	20
103	CLang	15

=====

**Q = In Oracle** – How to insert **100 Records** into Emp Table?

Insert into Emp (EmpID,EmpName,EmpSal) values (101,RRRS,10);

Insert into Emp (EmpID,EmpName,EmpSal) values (102,Chenna,20);

Insert into Emp (EmpID,EmpName,EmpSal) values (103,CLang,15);

=====100 times

In “**SAP Comm**” –

Insert\_Update Emp;EmpID[unique=true];EmpName;EmpSal

;101;RRRS;10

;102;Chenna;20

;103;CLang;15

=====

**Q =** What is the general **syntax** of the **ImpEx**?

**Mode** Itemtype;att1[m1=v1,m2=v2];att2[m1=v1][m2=v2];att3;===

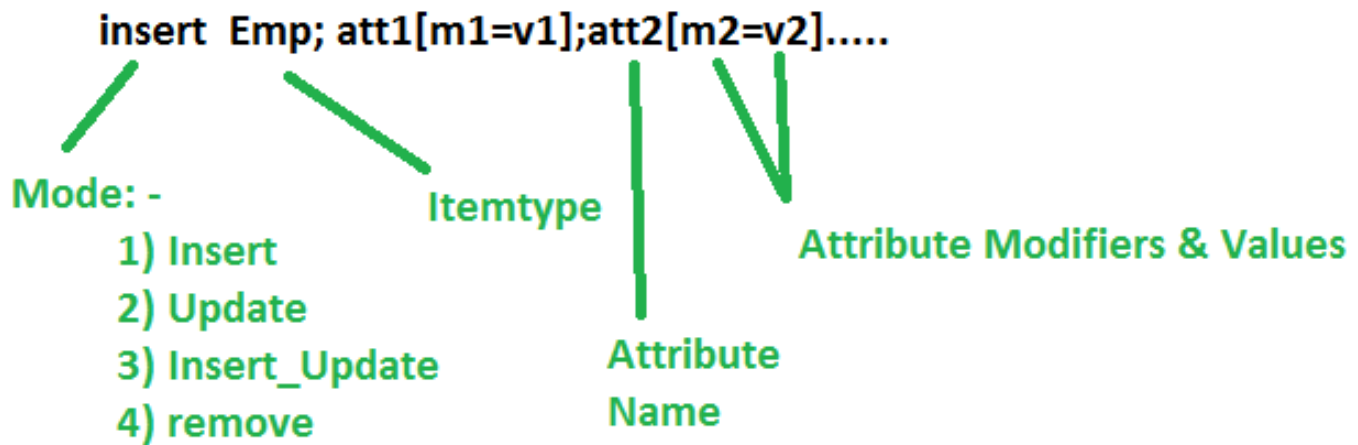
;Record 1

;Record 2

=====

**Note** = "SAP Comm" allows us to change / use **delimiters** = ; / # / , / ...

**Q** = What are the Terminologies (or) Jargons used in ImpEx?



**Q** = How can we specify multiple modifiers for attribute?

1st Method = att1[m1=v1, m2=v2, ....];

2nd Method = att1[m1=v1][m2=v2]...;

**Q: Why do we use Macro's in C Lang?**

We can define the macros in C Lang with #define.

Let's say -- Your Application is support 2 Lang (en & fr).

```
main()
{
    en & fr;
    ===
    en & fr;
    ===
    fun(en & fr);
}
fun(en & fr)
{
    en & fr;
    ===
}
```

This is what we have Today

Tomorrow -- Your manager came & asking you to add another Lang = de.

So now, you need to change in 40 places.

**Q: Is there any better way to do this?**

```
#define lang="en&fr";
```

```
main()
```

```
{
```

```
    lang;
```

```
    ===
```

```
    lang;
```

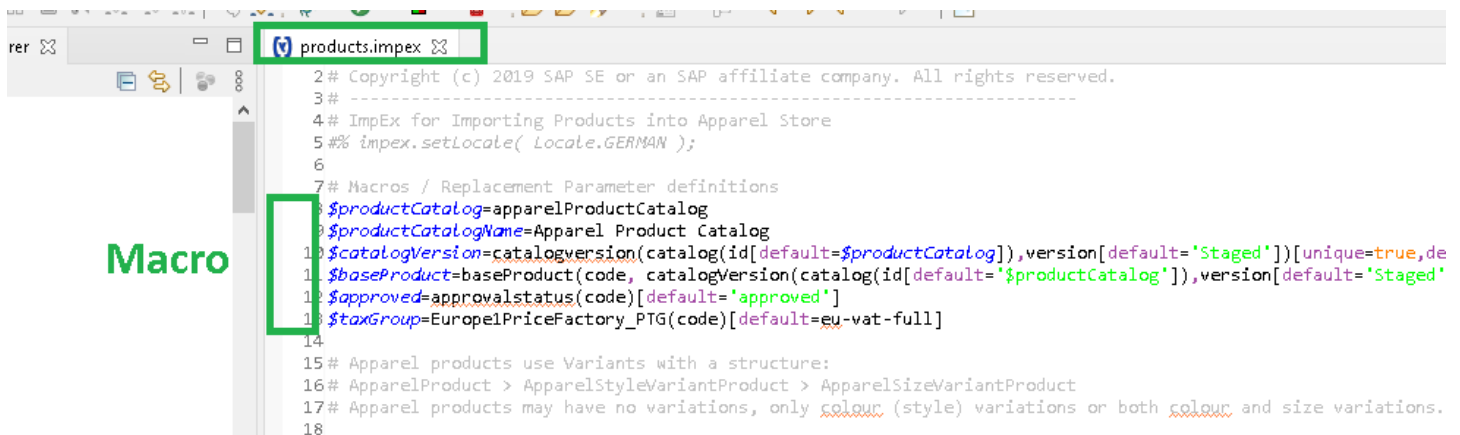
```
    ===
```

```
}
```

If we go with this. Tomorrow if your manager is asking you to add another Lang = de.  
No -- You will change only in 1 place.

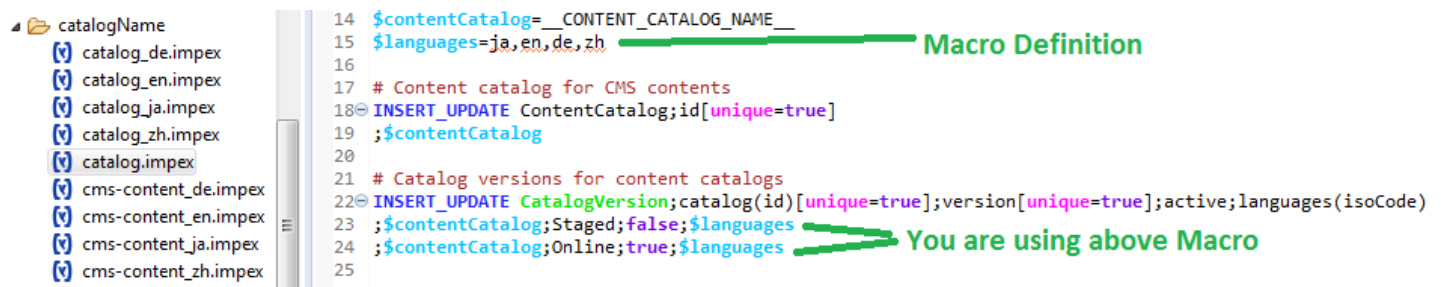
**Note** = “SAP Comm” ImpEx’s uses the Macros.

In C Lang - #define ===== In “SAP Comm” - \$===



The screenshot shows the SAP ImpEx editor with the file 'products.impex' open. A green box highlights the file name in the Project Explorer. Another green box highlights the macro definitions section of the code. The word 'Macro' is written in green to the left of the code. The code includes a copyright notice, a comment about the purpose of the ImpEx, and several macro definitions for product catalog, version, base product, approval status, and tax group.

```
2 # Copyright (c) 2019 SAP SE or an SAP affiliate company. All rights reserved.
3 #
4 # ImpEx for Importing Products into Apparel Store
5 %% impex.setLocale( Locale.GERMAN );
6
7 # Macros / Replacement Parameter definitions
8 $productCatalog=apparelProductCatalog
9 $productCatalogName=Apparel Product Catalog
10 $catalogVersion=catalogversion(catalog(id[default=$productCatalog]),version[default='Staged'])[unique=true,de
11 $baseProduct=baseProduct(code, catalogversion(catalog(id[default=$productCatalog]),version[default='Staged'
12 $approved=approvalstatus(code)[default='approved']
13 $taxGroup=Europe1PriceFactory_PTG(code)[default=eu-vat-full]
14
15 # Apparel products use Variants with a structure:
16 # ApparelProduct > ApparelStyleVariantProduct > ApparelSizeVariantProduct
17 # Apparel products may have no variations, only colour (style) variations or both colour and size variations.
18
```



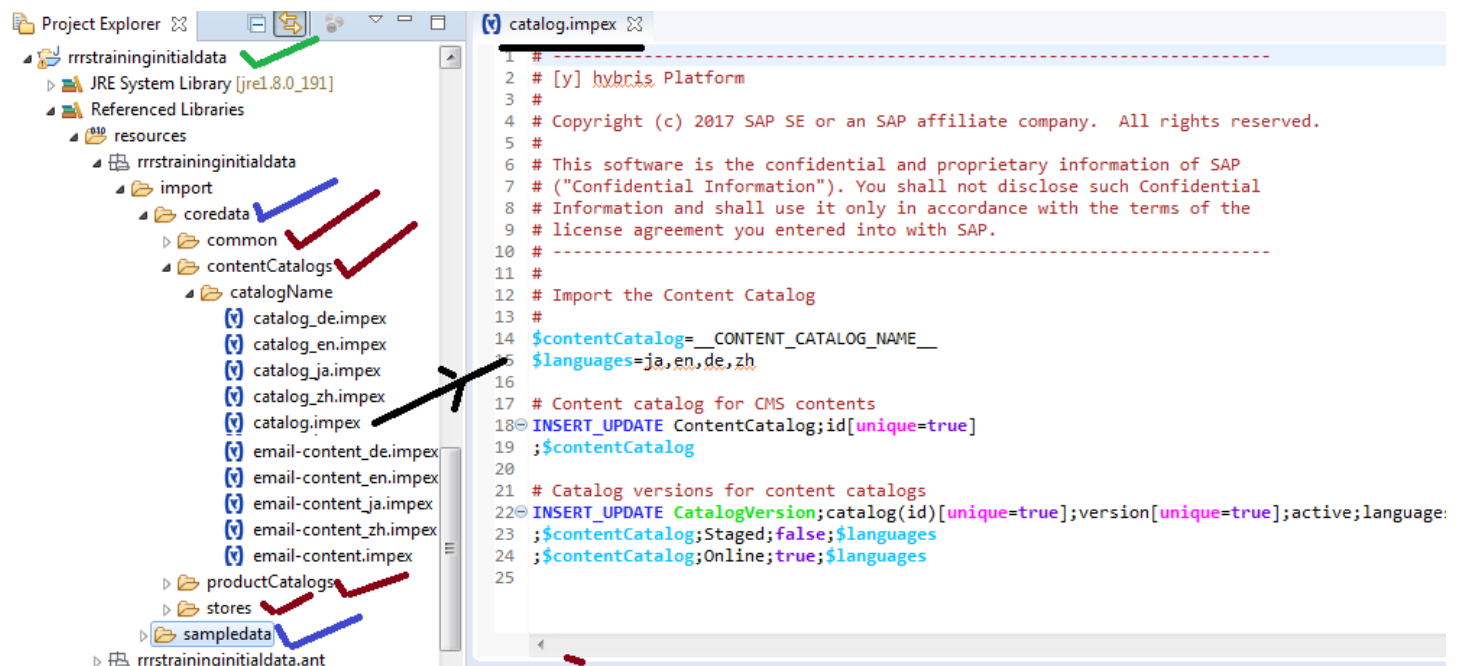
The screenshot shows the SAP ImpEx editor with the file 'catalogName.impex' open. The code defines a macro for content catalog and then uses it in the main ImpEx statement. Green annotations highlight the macro definition and its usage.

```
14 $contentCatalog=__CONTENT_CATALOG_NAME__
15 $languages=ja,en,de,zh
16
17 # Content catalog for CMS contents
18 INSERT_UPDATE ContentCatalog;id[unique=true]
19 ;$contentCatalog
20
21 # Catalog versions for content catalogs
22 INSERT_UPDATE CatalogVersion;catalog(id)[unique=true];version[unique=true];active;languages(isoCode)
23 ;$contentCatalog;Staged;false;$languages
24 ;$contentCatalog;Online;true;$languages
25
```

Macro Definition

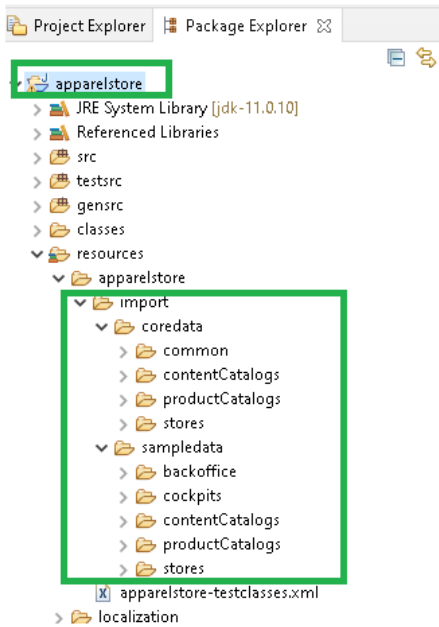
You are using above Macro

**Note:** Generally – We write the ImpEx “\*Initialdata” Ext.



The screenshot shows the SAP ImpEx editor with the file 'catalogName.impex' open. The Project Explorer on the left shows the file structure, including the 'initialdata' folder. A green checkmark is placed next to the 'initialdata' folder. A red checkmark is placed next to the 'catalogName.impex' file. A black arrow points from the 'catalogName.impex' file in the Project Explorer to the code in the editor.

```
1 #
2 # [y] hybris Platform
3 #
4 # Copyright (c) 2017 SAP SE or an SAP affiliate company. All rights reserved.
5 #
6 # This software is the confidential and proprietary information of SAP
7 # ("Confidential Information"). You shall not disclose such Confidential
8 # Information and shall use it only in accordance with the terms of the
9 # license agreement you entered into with SAP.
10 #
11 #
12 # Import the Content Catalog
13 #
14 $contentCatalog=__CONTENT_CATALOG_NAME__
15 $languages=ja,en,de,zh
16
17 # Content catalog for CMS contents
18 INSERT_UPDATE ContentCatalog;id[unique=true]
19 ;$contentCatalog
20
21 # Catalog versions for content catalogs
22 INSERT_UPDATE CatalogVersion;catalog(id)[unique=true];version[unique=true];active;language:
23 ;$contentCatalog;Staged;false;$languages
24 ;$contentCatalog;Online;true;$languages
25
```



Name	Date modified	Type	Size
coredata	26-05-2021 20:24	File folder	
sampledata	26-05-2021 20:24	File folder	

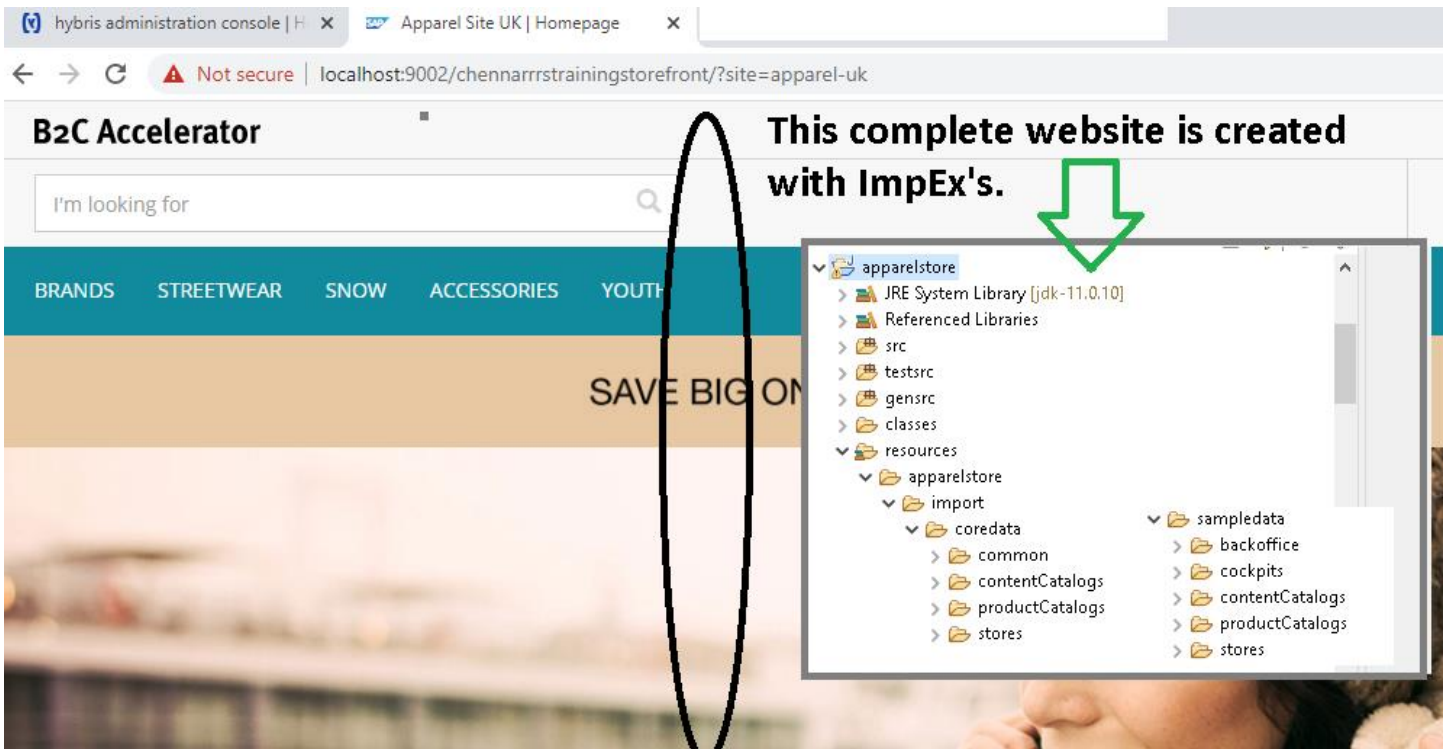
Mainly – We have 2 types of Data: -

### 1) CoreData =

- a. **Common** = essential-data.impex (Language ... Currency ... Titles ... )
- b. **ContentCatalogs** = catalog.impex ... cms-content.impex ... email ...
- c. **ProductCatalogs** =
- d. **Store** = site.impex .... Store.impex .... solr.impex ...

### 2) SampleData =

- a. **backoffice**
- b. **cockpits**
- c. **contentCatalogs**
- d. **productCatalogs**
- e. **stores**





These ImpEx are required to Setup this Site.

Q: How these ImpEx are executed? = During INIT / Update

Q: Where is the Logic says that ... What to Execute during INIT & What to execute during Update?

```

65  /* This method will be called by system creator during initialization and system update */
66  @SystemSetup(type = Type.ESSENTIAL, process = Process.ALL)
67  public void createEssentialData(final SystemSetupContext context)
68  {
69      // Add Essential Data here as you require
70  }
71
72  /* This method will be called during the system initialization. */
73  @SystemSetup(type = Type.PROJECT, process = Process.ALL)
74  public void createProjectData(final SystemSetupContext context)
75  {
76      // Add import data for each site you have configured
77  }
  
```

Q = What are the different ways to execute the ImpEx?

Sol1 = hAC

hybris administration console

You're Administrator [logout](#)

Platform Monitoring Maintenance Console

Scripting Languages FlexibleSearch **ImpEx Import** ImpEx Export LDAP

Import content Import script

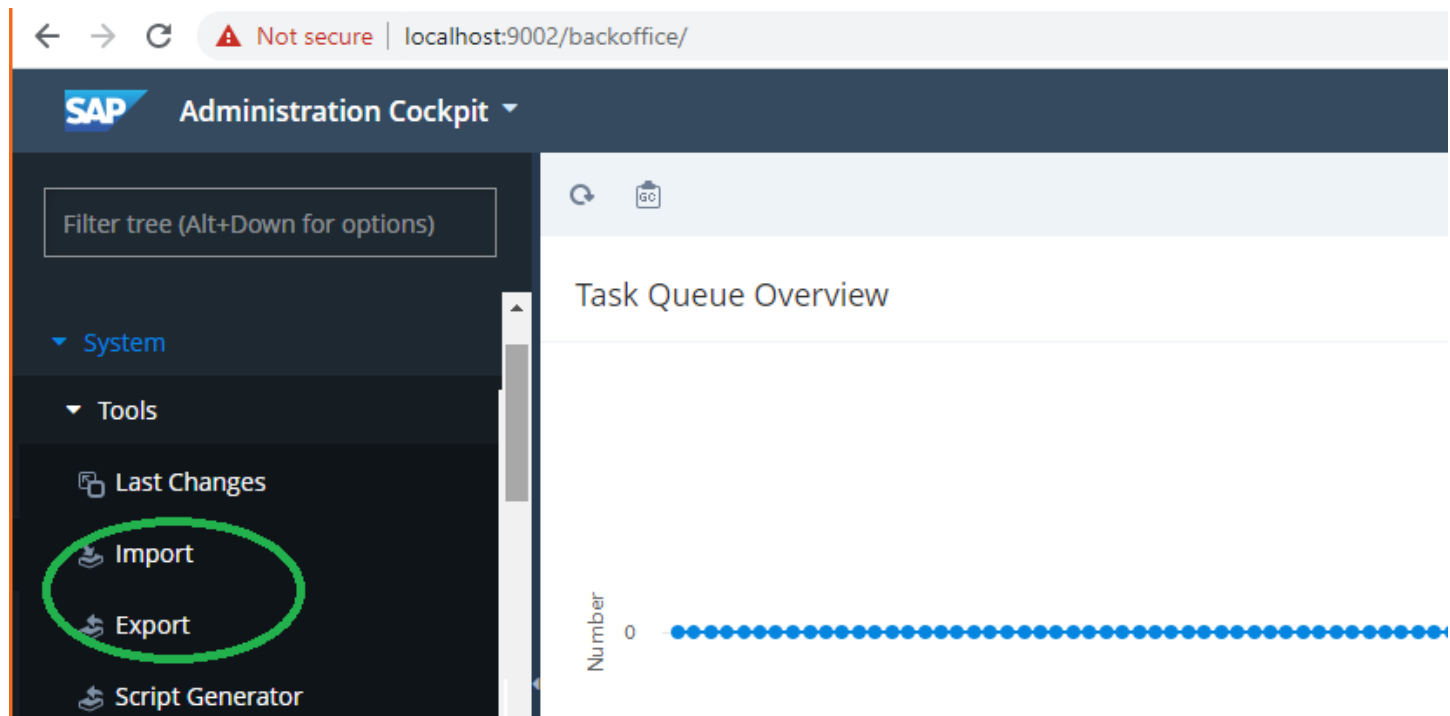
**Import content**

```

1  # Language
2  $lang=en
3
4  # Create CMS Site
5  UPDATE CMSSite;uid[unique=true];name[lang=$lang];locale[lang=$lang]
6  ;apparel-uk;"DRL Site UK";en_GB
7
  
```

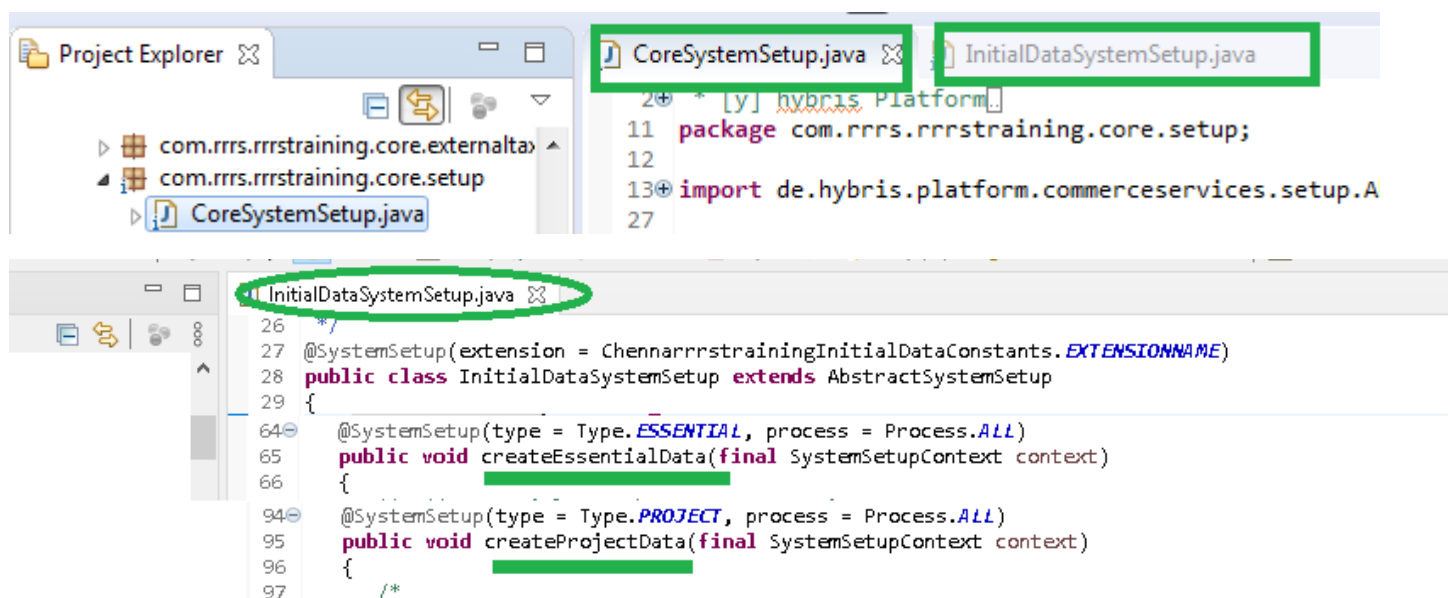
[Clear content](#) [Import content](#) [Validate content](#)

## Sol2 = Backoffice

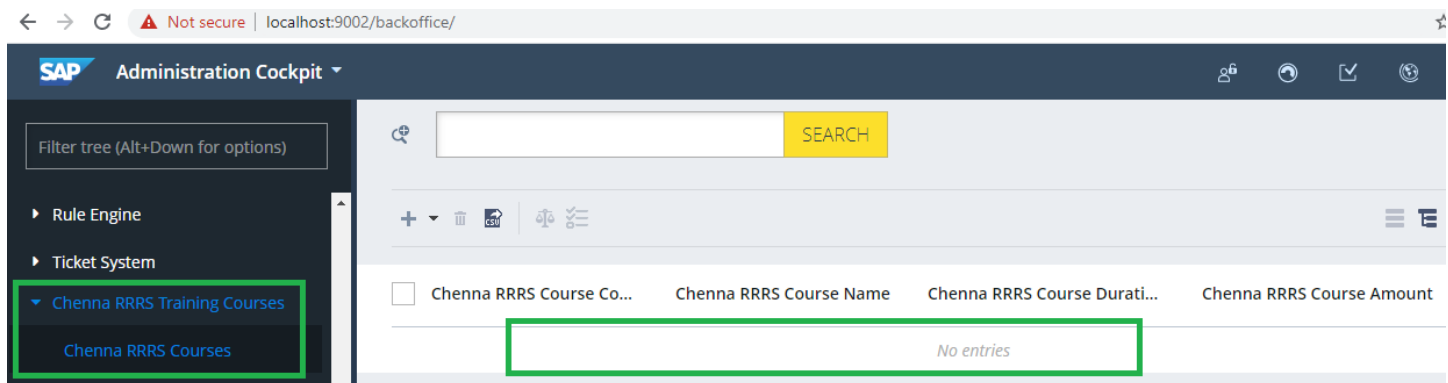
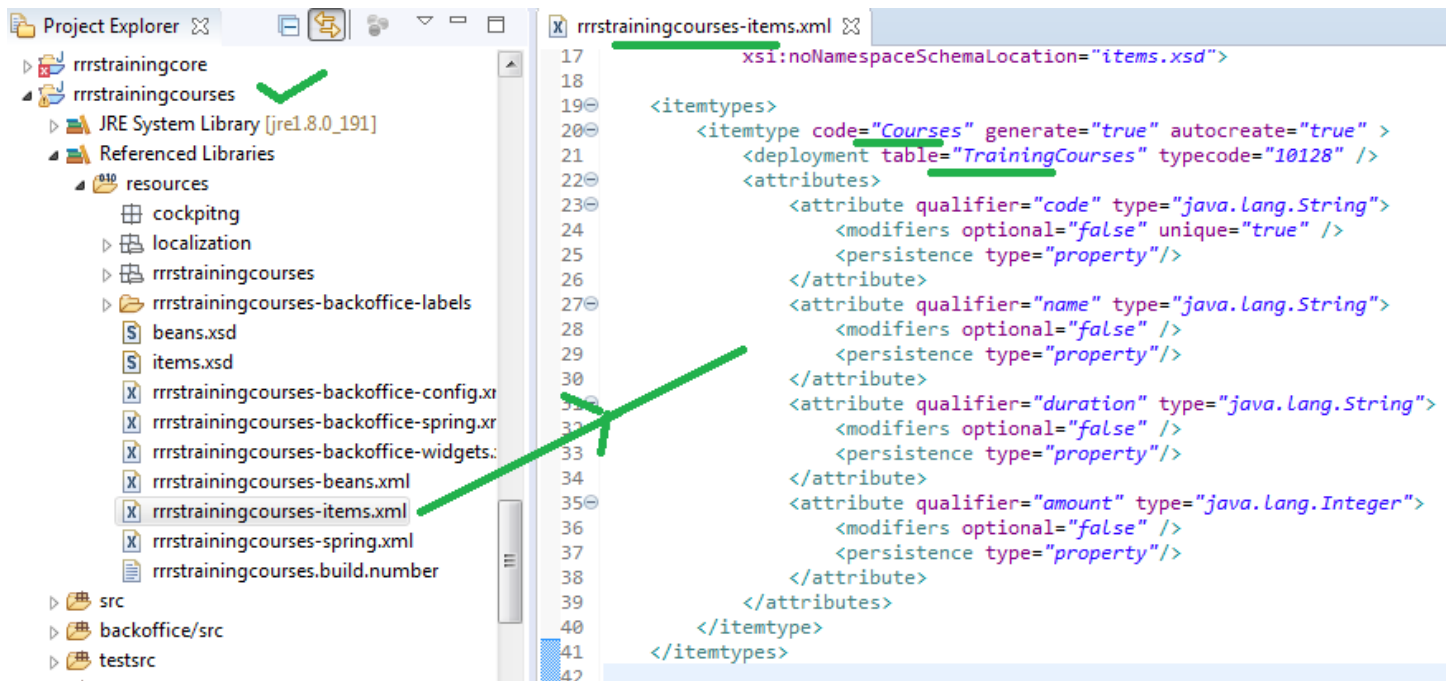


## Sol3 = Programmatic way / ImpEx == During INIT / Update time.

```
INSERT_UPDATE Title;code[unique=true];name[lang=zh];name[lang=en];name[lang=ja];name[lang=pt]
;0001 ;Ms. ;Ms. ;æ$~ ;Ms.
;0002 ;Mr. ;Mr. ;æ$~ ;Mr.
```

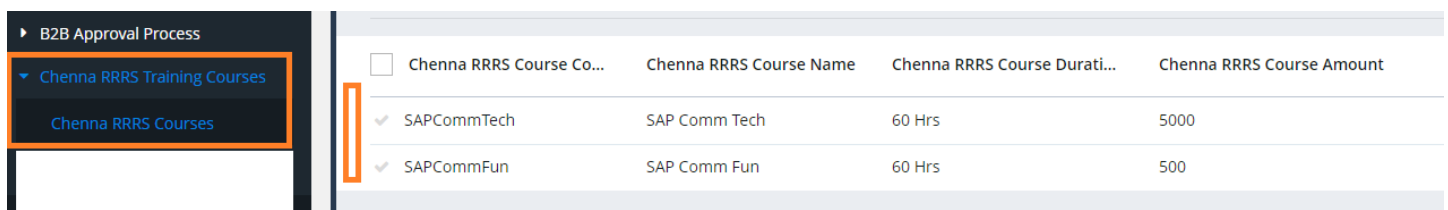


## Example =

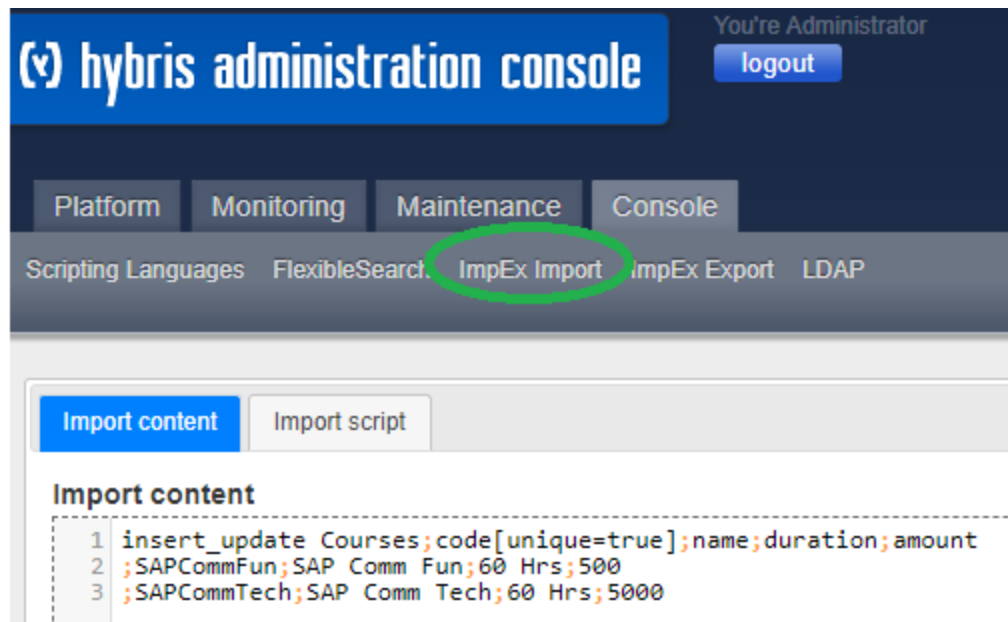


Q1 = How to insert few records into above table (or) itemtype?

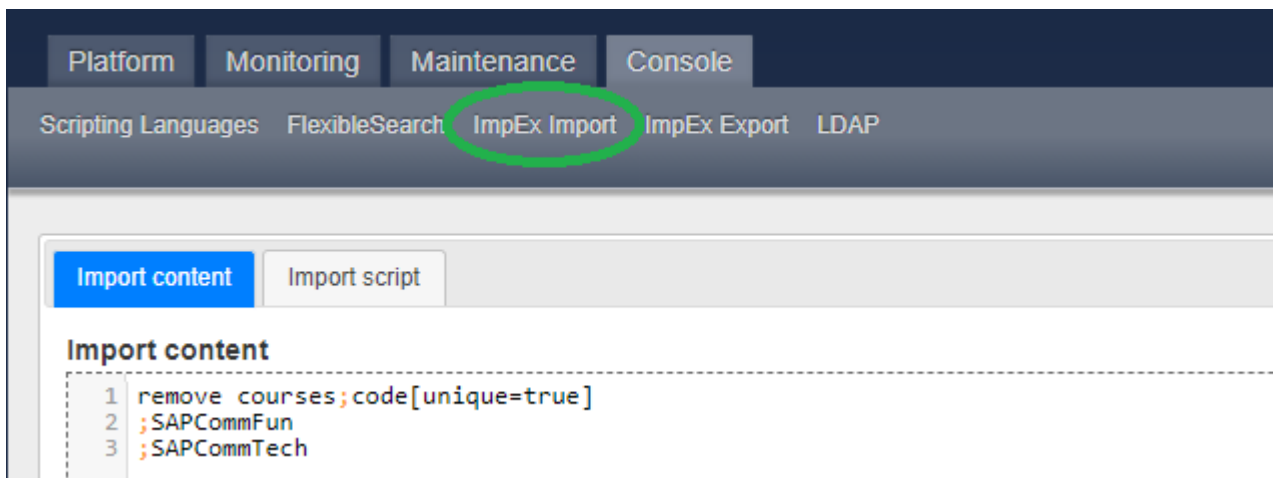
Results =



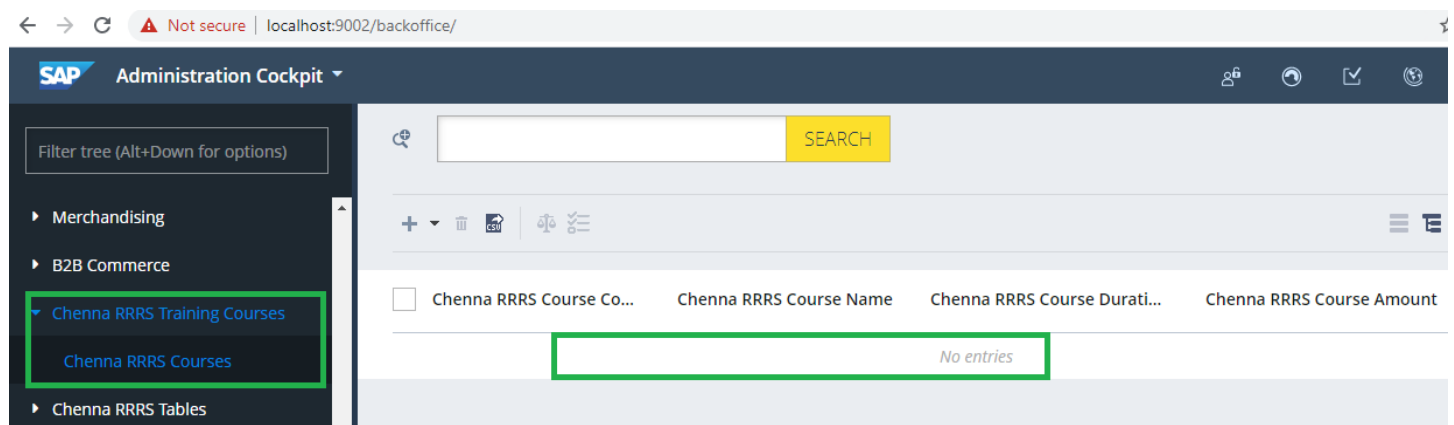
ImpEx =



Q2 = How to remove few records?



Results =



Contact Us = [ChennaReddyTraining@RRRS.CO.IN](mailto:ChennaReddyTraining@RRRS.CO.IN)

**Scenario** = Let's say ... We have 20K records coming from SAP system & you are loading those.

But 1621 and 1881 records is having error.

Q = What happens in this case.

- 1) All records will not be processed
- 2) **Only 1621 & 1881 records are not processed**
- 3) Upto 1620 record is processed
- 4) All above?

The screenshot shows the SAP ImpEx Import interface. The 'Import content' tab is selected. The 'Import content' section displays a list of records. Records 103 and 105 are highlighted with red arrows and text indicating they have wrong data.

Line	Record
1	insert_update Courses;code[unique=true];name;duration;amount
2	;101;Chenna;120 Hrs;500
3	;102;RRRS;120 Hrs;500
4	;103;Chenna1;110 Hrs;abc
5	;104;RRRS1;120 Hrs;550
6	;105;Chena2;110 Hrs;hello
7	;106;SAPComm;120 Hrs;600
8	

**These 2 records are having wrong data**

 **Failed Record details**

insert\_update Courses;code[unique=true];name;duration;amount  
",,,cannot parse number 'abc' with format specified pattern '#,##0' due to Unparseable number: ""abc"";103;Chenna1;110 Hrs;abc  
",,,cannot parse number 'hello' with format specified pattern '#,##0' due to Unparseable number: ""hello"";105;Chena2;110 Hrs;hello

**Results =**

The screenshot shows the SAP B2B Commerce interface. The 'Chenna RRRS Training Courses' section is highlighted. The table below shows the imported records.

	Chenna RRRS Course Co...	Chenna RRRS Course Name	Chenna RRRS Course Durati...	Chenna RRRS Course Amount
✓ 106		SAPComm	120 Hrs	600
✓ 104		RRRS1	120 Hrs	550
✓ 102		RRRS	120 Hrs	500
✓ 101		Chenna	120 Hrs	500

## How to see the failed record details?

```
33922 | at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49) [tomcat-coyote.jar:8.5.57]
33923 | at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128) [?:?]
33924 | at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628) [?:?]
33925 | at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61) [tomcat-util.jar:8.5.57]
33926 | at java.lang.Thread.run(Thread.java:834) [?:?]
33927 | Caused by: de.hybris.platform.jalo.JaloInvalidParameterException: cannot parse number 'BCD' with format specified pa
33928 | at de.hybris.platform.impex.jalo.translators.AtomicValueTranslator.convertToJalo(AtomicValueTranslator.java:599)
33929 | at de.hybris.platform.impex.jalo.translators.SingleValueTranslator.importValue(SingleValueTranslator.java:49) ~[
33930 | at de.hybris.platform.impex.jalo.header.StandardColumnDescriptor.importValue(StandardColumnDescriptor.java:492)
33931 | at de.hybris.platform.impex.jalo.imp.DefaultValueLineTranslator.translateColumnValues(DefaultValueLineTranslator
33932 | at de.hybris.platform.impex.jalo.imp.DefaultImportProcessor.processInsertLine(DefaultImportProcessor.java:540) ~
33933 | at de.hybris.platform.impex.jalo.imp.DefaultImportProcessor.processInsertUpdateLineInternal(DefaultImportProcess
33934 | at de.hybris.platform.impex.jalo.imp.DefaultImportProcessor$4.execute(DefaultImportProcessor.java:465) ~[impexse
33935 | at de.hybris.platform.impex.jalo.imp.DefaultImportProcessor$4.execute(DefaultImportProcessor.java:1) ~[impexserv
33936 | at de.hybris.platform.tx.Transaction.execute(Transaction.java:1174) ~[coreserver.jar:?]
33937 | at de.hybris.platform.tx.Transaction.execute(Transaction.java:1142) ~[coreserver.jar:?]
33938 | at de.hybris.platform.impex.jalo.imp.DefaultImportProcessor.processInsertUpdateLine(DefaultImportProcessor.java:
33939 | ... 147 more
33940 | INFO [hybrisHTTP14] (00000E08) [Importer] Finished 2 pass in 0d 00h:00m:00s:059ms - processed: 2, dumped: 2 (last p
33941 | ERROR [hybrisHTTP14] (00000E08) [CronJobErrorHandler] de.hybris.platform.impex.jalo.ImpExException: Can not resolve
33942 | any more lines ... Aborting further passes (at pass 2). Finally could not import 2 lines! [HY-123]
33943 | WARN [hybrisHTTP14] (00000E08) [Importer] Import finished with errors within 0d 00h:00m:00s:124ms
33944 | ERROR [hybrisHTTP14] [DefaultImportService] Import has caused an error, see logs of cronjob with code=00000E08 for f
33945
```

(OR)

Administration Cockpit

Search: [ ] SEARCH

1 / 10 480 items

Code	Job definition	Current stat...	Last result
00000E08	ImpEx-Import	FINISHED	SYSTEM ERR...

0 ITEMS SELECTED

ImpEx-Import : 00000E08 - FINISHED - FAILURE

REFRESH SAVE

LOG TASK RUN AS TIME SCHEDULE SYSTEM RECOVERY ADMINISTRATION

Download This

Files Days Old 14

Log files

20210710\_032501-a64cd19c-9aca-479... x

== Download the Log file

Downloads >

Name	Date modified	Type	Size
Today (2)			
20210710_032501-a64cd19c-9aca-4796-ad7e-c6...			
20210710_032501-a64cd19c-9aca-4796-ad7e-c6...			

Unzip This

Unzipped Results & Open this

Downloads > 20210710\_032501-a64cd19c-9aca-4796-ad7e-c690979f3a04

8796716171765

C:\Users\Del\Downloads\20210710\_032501-a64cd19c-9aca-4796-ad7e-c690979f3a04\8796716171765.log - Notepad++

```

1 Starting ImpEx cronjob "ImpEx-Import"
2 Exception occurred, will ignore: de.hybris.platform.impex.jalo.ImpExException: cannot parse number 'ABC' with format s
3 Exception occurred, will ignore: de.hybris.platform.impex.jalo.ImpExException: cannot parse number 'BCD' with format s
4 Finished 1 pass in 0d 00h:00m:00s:064ms - processed: 6, dumped: 2 (last pass: 0)
5 Starting pass 2
6 dumped unresolved line ValueLine[unresolvable:cannot parse number 'ABC' with format specified pattern '#,##0' due to
7 Exception occurred, will ignore: de.hybris.platform.impex.jalo.ImpExException: cannot parse number 'ABC' with format s
8 dumped unresolved line ValueLine[unresolvable:cannot parse number 'BCD' with format specified pattern '#,##0' due to
9 Exception occurred, will ignore: de.hybris.platform.impex.jalo.ImpExException: cannot parse number 'BCD' with format s
10 Finished 2 pass in 0d 00h:00m:00s:059ms - processed: 2, dumped: 2 (last pass: 2)
11 de.hybris.platform.impex.jalo.ImpExException: Can not resolve any more lines ... Aborting further passes (at pass 2).
12 Import finished with errors within 0d 00h:00m:00s:174ms
13

```

==== We can see the Job details as shown in below screenshot: -

Background Processes

CronJobs

Code	Job definition	Current stat...	Last result	Timeta
00000EO8	ImpEx-Import	FINISHED	SYSTEM ERR...	Not sch

0 ITEMS SELECTED

ImpEx-Import : 00000EO8 - FINISHED - FAILURE

LOG TASK RUN AS TIME SCHEDULE SYSTEM RECOVERY ADMINISTRATION

Code: 00000EO8

Current status: FINISHED

Job: ImpEx-Import

Last result: SYSTEM ERROR

Timetable: Not scheduled

Last start time: Jul 10, 2021 3:39:37 AM

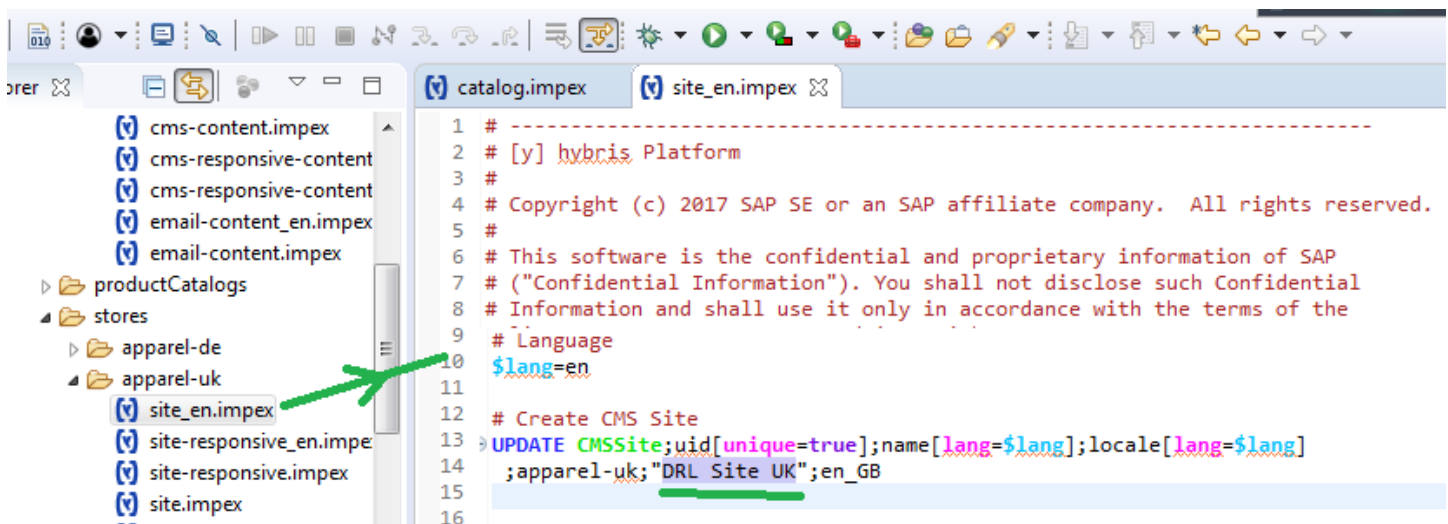
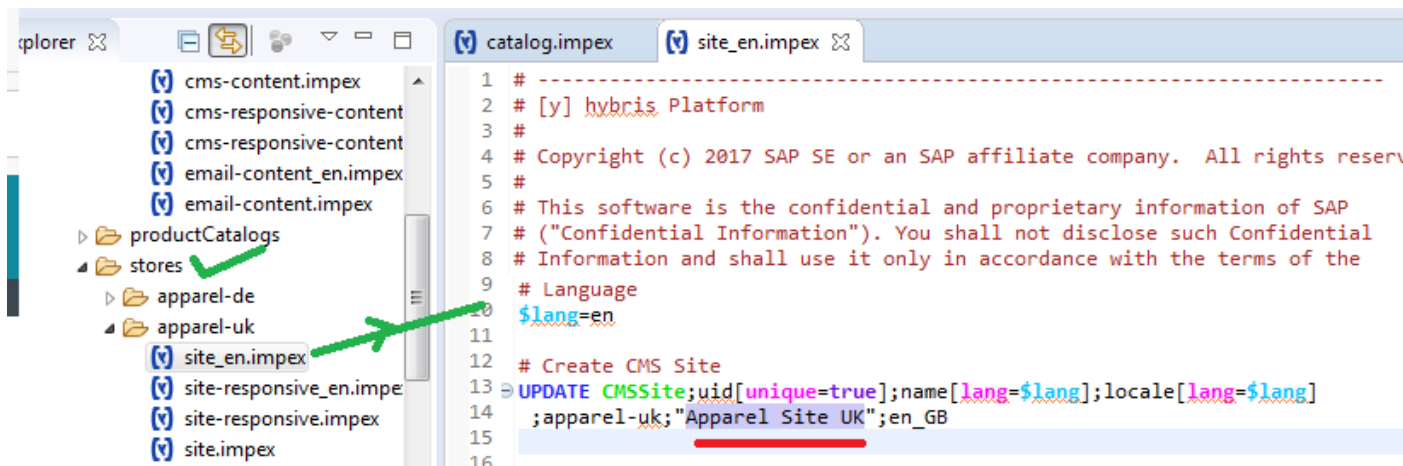
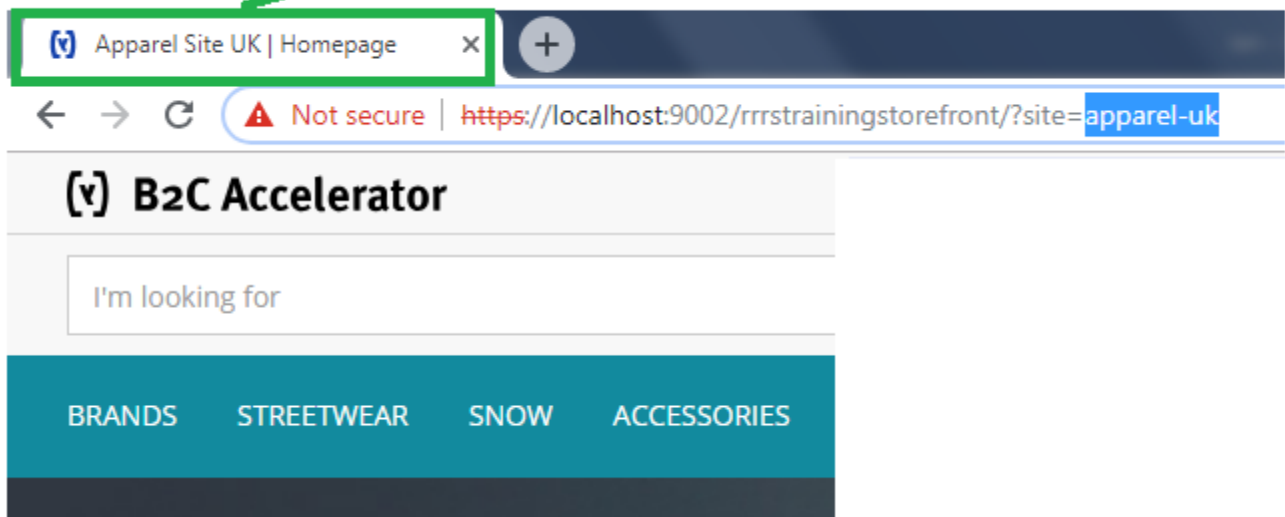
Enabled: True

Last end time: Jul 10, 2021 3:39:37 AM

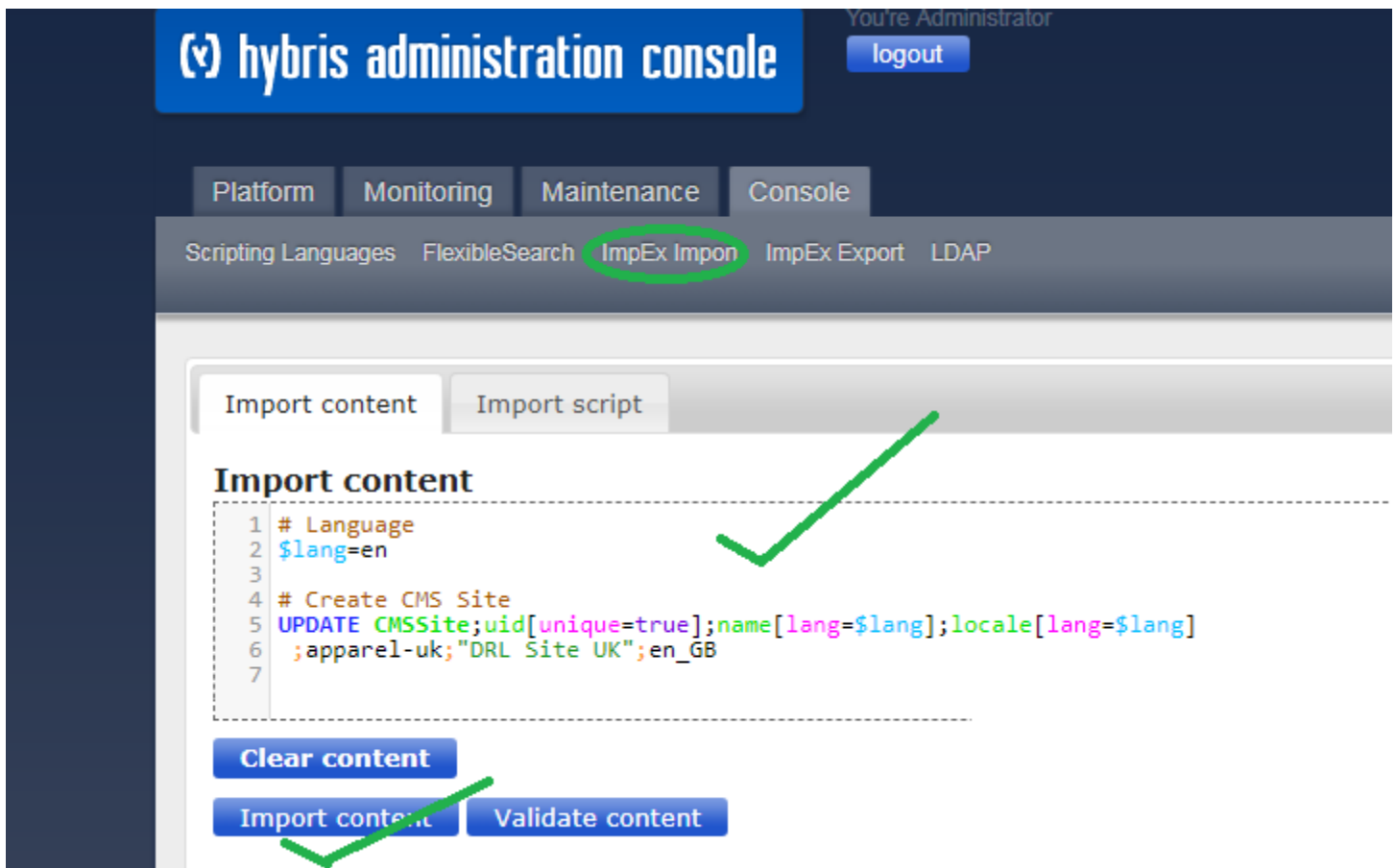


## Requirement = How to change the Site Title?

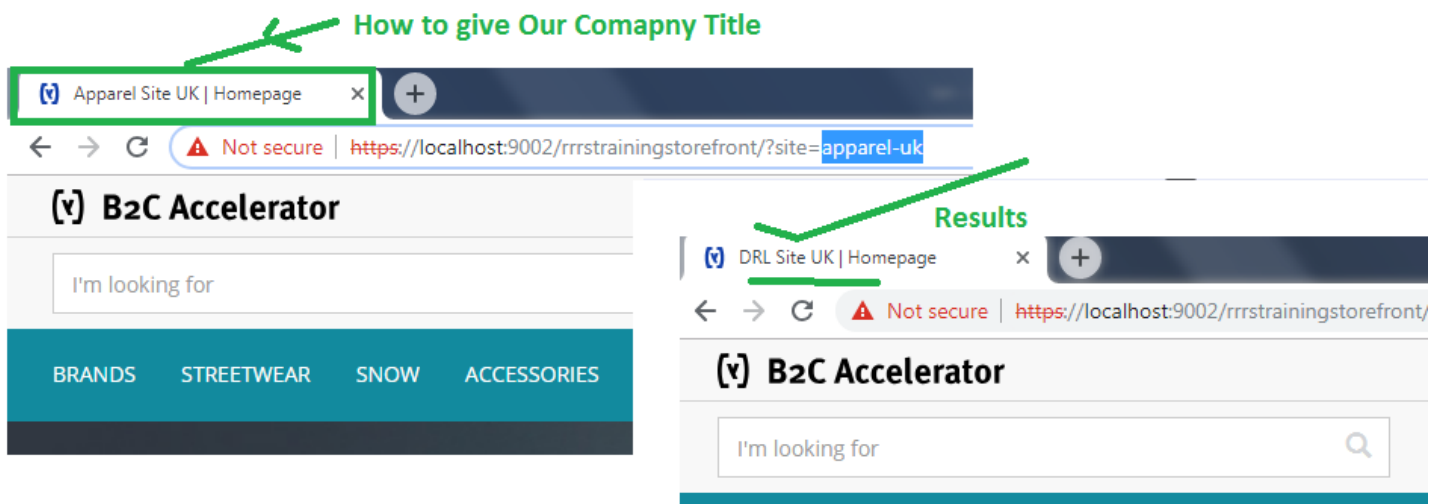
How to give Our Company Title







Results =



**Q = How to import the ImpEx which is already there in file?**

hybris administration console

logout

Platform Monitoring Maintenance Console

Scripting Languages FlexibleSearch **ImpEx Import** ImpEx Export LDAP

Import content Import script

**Import script**

Choose file ✓

Script encoding UTF-8

Max. threads 1

Import validation mode import\_strict ▼

Legacy mode ☐

Enable code execution ☐

Distributed mode ☐

Direct persistence ☐

Import file

**Example – Import below ImpEx: -**

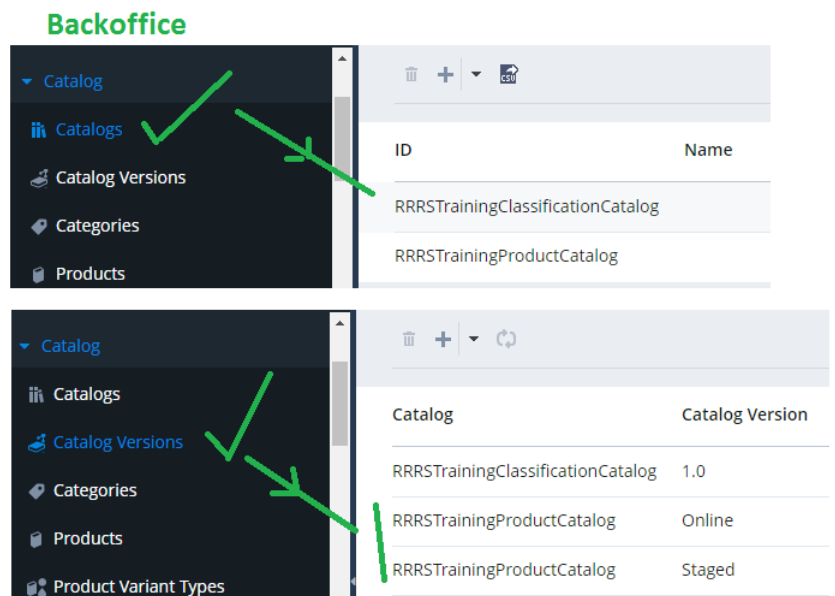
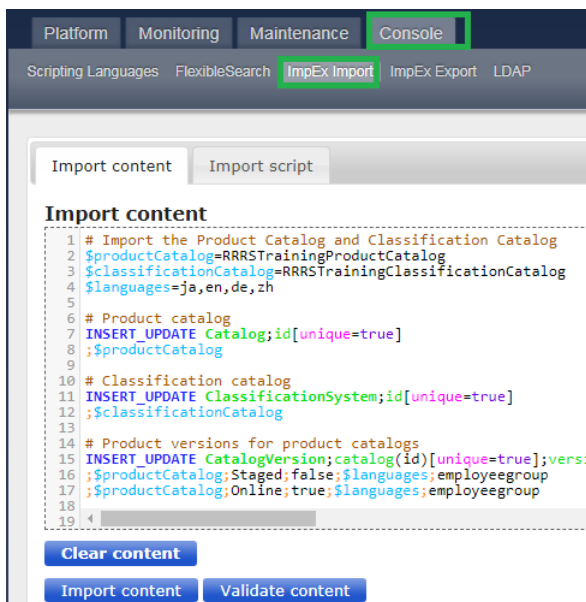
```
# Import the Product Catalog and Classification Catalog
$productCatalog=RRRSTrainingProductCatalog
$classificationCatalog=RRRSTrainingClassificationCatalog
$languages=ja,en,de,zh

# Product catalog
INSERT_UPDATE Catalog;id[unique=true]
;$productCatalog

# Classification catalog
INSERT_UPDATE ClassificationSystem;id[unique=true]
;$classificationCatalog

# Product versions for product catalogs
INSERT_UPDATE CatalogVersion;catalog(id) [unique=true];version[unique=true];active;languages(isoCode);readPrincipals(uid)
;$productCatalog;Staged;false;$languages;employeeegroup
;$productCatalog;Online;true;$languages;employeeegroup

# Insert Classifications System Version
INSERT_UPDATE ClassificationSystemVersion;catalog(id) [unique=true];version[unique=true];active;inclPacking[virtual=true,default=true];
inclDuty[virtual=true,default=true];inclFreight[virtual=true,default=true];inclAssurance[virtual=true,default=true]
;$classificationCatalog;1.0;true
```



**Business Scenario** = Assume that Business **did not write ImpEx** to insert Records into **Courses**.

Business Entered the **Records directly from Backoffice [Dev System]**.

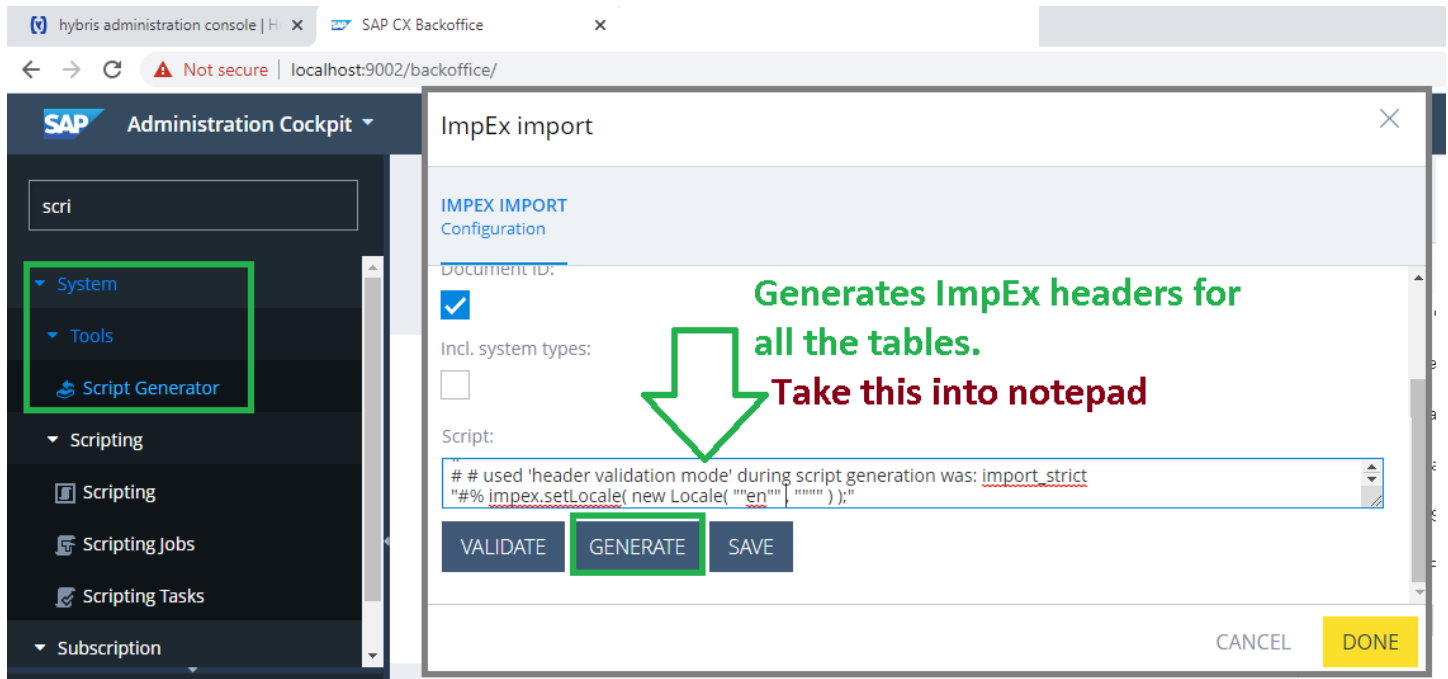


**Q1** = How to move above entered records into other Environments (QA ... UAT ... Prod). (or)

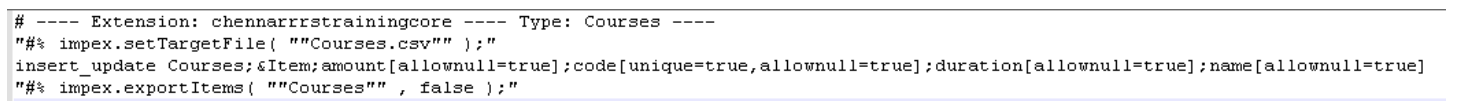
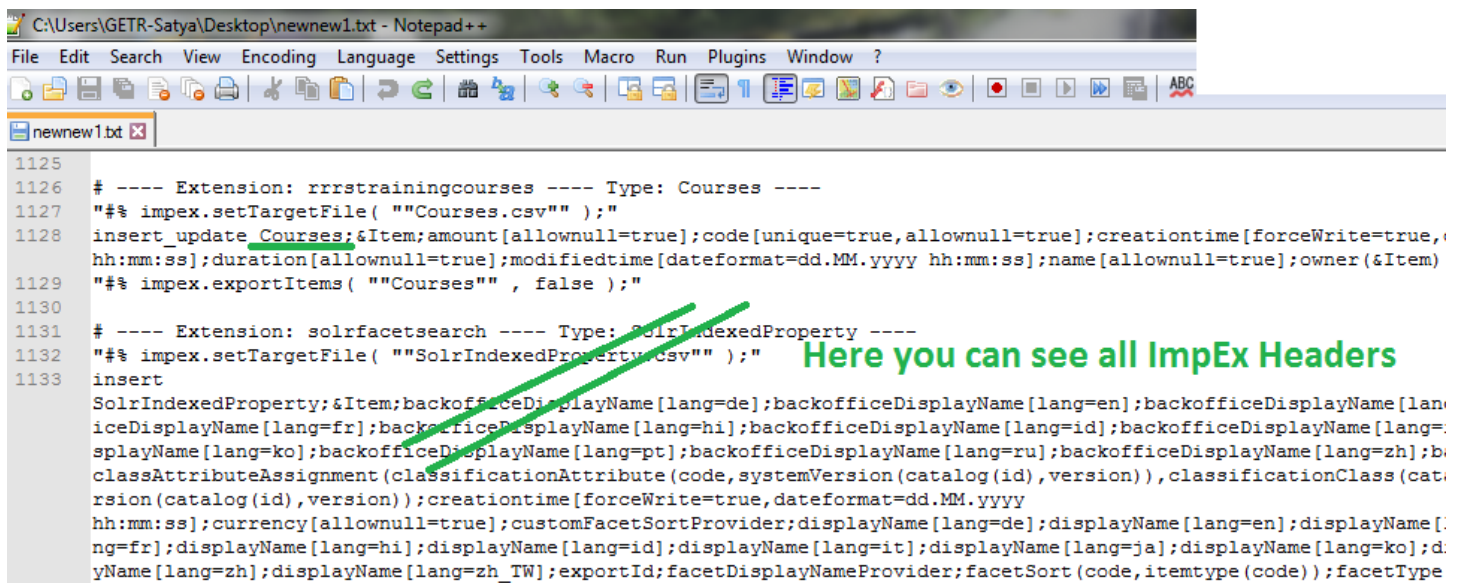
**Q2** = Sometimes, it's difficult to write the ImpEx headers. Is there any way you can download (or) check (or) take the ImpEx headers? (or)

**Q3** = How to use the Script Generators?

## Step 1 = Backoffice – Script Generator

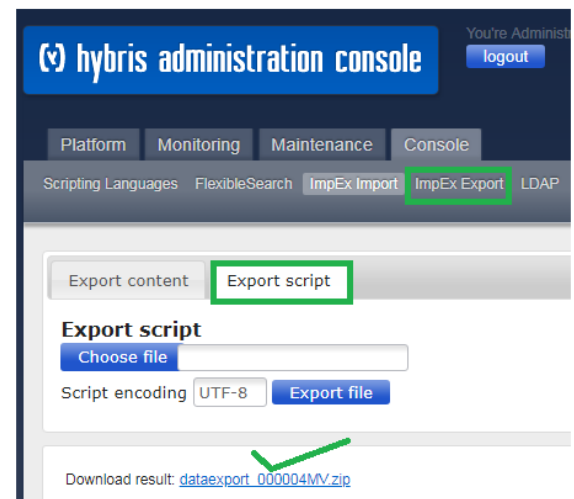
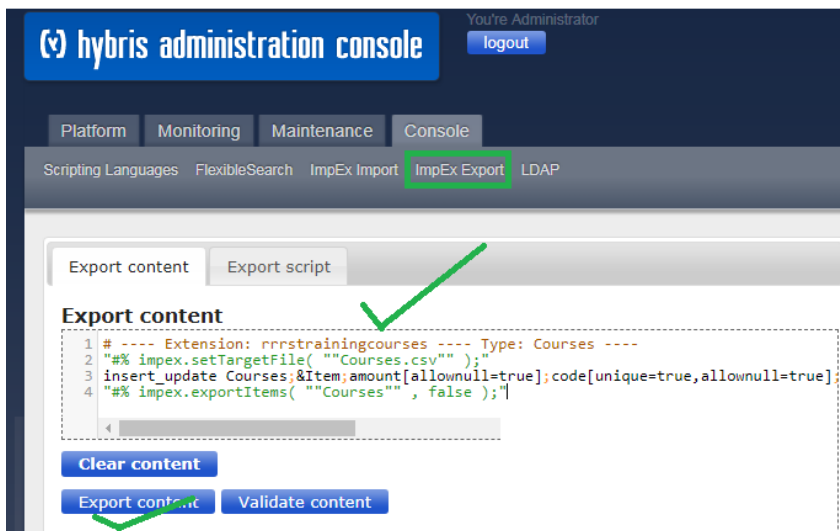


## Step 2 = Take the Generated ImpEx headers into Notepad

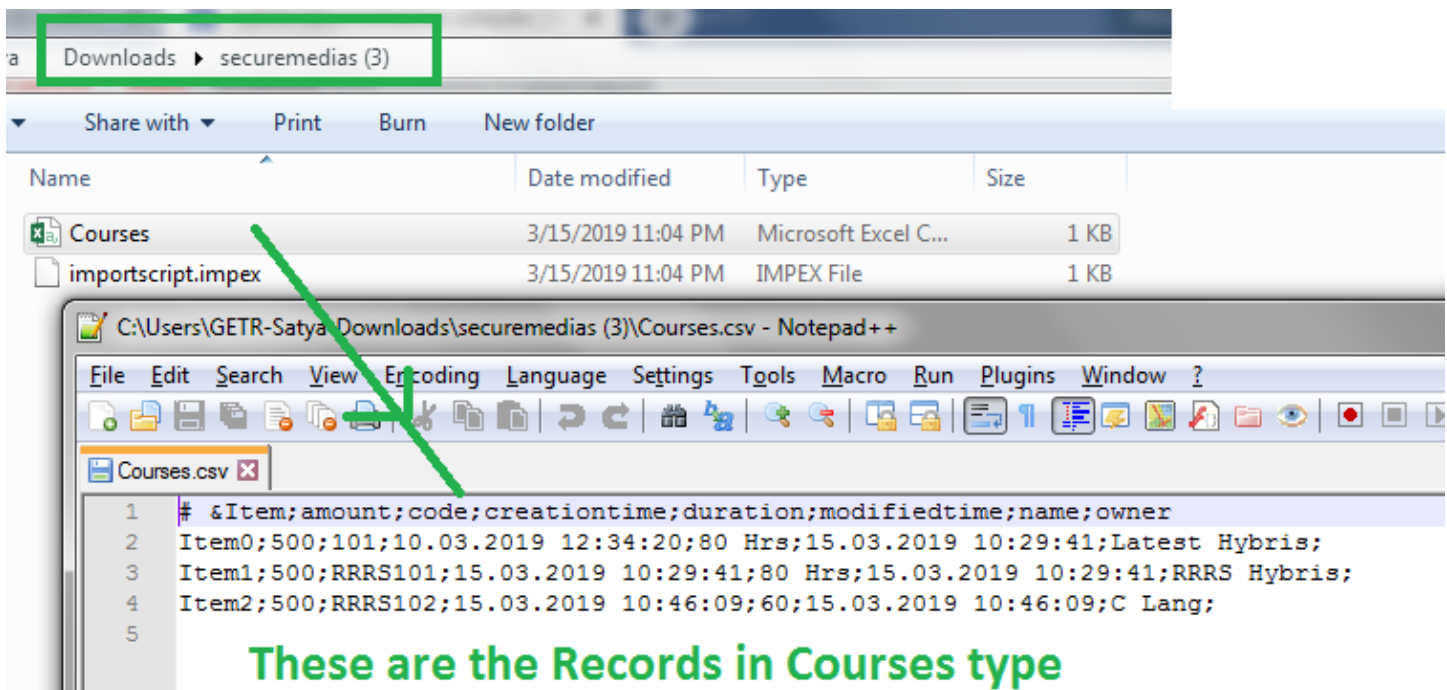


**ImpEx Header for Courses**

**Step 3 = Take the Courses ImpEx Header & fetch (or) get the Records from source Env [Dev System].**



Unzip the download file.



Downloads > securemedias (3)

es

Name

Courses

importscript.impex

AutoSave Off

Courses - Read-Only - Excel (Unlicensed Product)

File Home Insert Page Layout Formulas Data Review View Developer Help

Paste

Clipboard

Font

Alignment

Number

Styles

PRODUCT NOTICE Most of the features of Excel have been disabled because it hasn't been activated. [Activate](#)

A1

# &Item;amount;code;creationtime;duration;modifiedtime;name;owner

	A	B	C	D	E	F	G	H	I	J	K
1	# &Item;amount;code;creationtime;duration;modifiedtime;name;owner										
2	Item0;500;101;10.03.2019 12:34:20;80 Hrs;15.03.2019 10:29:41;Latest Hybris123;										
3	Item1;500;RRRS101;15.03.2019 10:29:41;80 Hrs;15.03.2019 10:29:41;RRRS Hybris123;										
4	Item2;500;RRRS102;15.03.2019 10:46:09;60;15.03.2019 10:46:09;C Lang;										

Change Something

Downloads > securemedias (3)

Files

Name

Courses

importscript.impex

Courses.csv

```

1 # &Item;amount;code;duration;name
2 Item0;500;101;80 Hrs;Latest Hybris123;
3 Item1;500;RRRS101;80 Hrs;RRRS Hybris123;
4 Item2;500;RRRS102;60;C Lang;
5 Item3;500;RRRS103;60;C Lang New;

```

Do Changes ...  
Add Record ...

Change ImpEx if required

```

1
2 #-----
3 insert_update
4 Courses;&Item;amount[allownull=true];code[unique=true,allownull=true];duration[allownull=true];name[allownull=true]
5 "## impex.includeExternalDataMedia ( "Courses.csv" , "UTF-8" , ',' , 1 , -1 );"

```

Downloads > dataexport\_000005EQ

Name

Courses

importscript.impex

Changed File

02-06-2021 04:21

Microsoft Excel C...

1 KB

dataexport\_000005EQ.zip - WinRAR

File Commands Tools Favorites Options Help

Add Extract To Test View Delete Find Wizard Info VirusScan Comment SFX

dataexport\_000005EQ.zip - ZIP archive, unpacked size 577 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
Courses.csv	237	144	Microsoft Excel Co...	02-06-2021 04:21	53155713
importscript.im...	340	163	IMPEX File	02-06-2021 04:19	32171CCB

Drag & Drop inside ZIP File

## Step 4 = Go to the Target Evn [UAT / Prod] & Import the Records.

hybris administration console

Platform Monitoring Maintenance Console

Scripting Languages FlexibleSearch **ImpEx Import** ImpEx Export LDAP

Import content **Import script**

Choose file securemedias (3).zip

Script encoding UTF-8

Max. threads 1

Import validation mode import\_strict

Legacy mode ☐

Enable code execution ☒

Distributed mode ☐

Direct persistence ☐

**Import file**

Inbox

- System
- Catalog
- RRRSTraining
  - Courses**
- Multimedia

Course Code	Course Name	Course Duration	Course Amount
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500
101	Latest Hybris123	80 Hrs	500

```
C:\Windows\System32\cmd.exe - hybrisserver.bat
Q
INFO [hybrisHTTP4] [DefaultExportService] Export was successful (using cronjob with code=000005EQ)
INFO [hybrisHTTP3] [DefaultImportService] Starting import synchronous using cronjob with code=000005ER
INFO [hybrisHTTP3] (000005ER) [ImpExImportJob] Starting ImpEx cronjob "ImpEx-Import"
INFO [hybrisHTTP3] (000005ER) [Importer] Finished 1 pass in 0d 00h:00m:00s:634ms - processed: 6, no 1
ines dumped (last pass 0)
INFO [hybrisHTTP3] [DefaultImportService] Import was successful (using cronjob with code=000005ER)
```

## Results =

B2B Commerce

- Chenna RRRS Training Courses**
- Chenna RRRS Courses
- Chenna RRRS Tables
- Personalization
- Subscription

SAVED QUERIES

No queries

<input type="checkbox"/>	Chenna RRRS Course Co...	Chenna RRRS Course Name	Chenna RRRS Course Durati...	Chenna RRRS Course Amount
<input checked="" type="checkbox"/>	Spartacus	SAP Spartacus	40 Hrs	600
<input checked="" type="checkbox"/>	CLang	C Language	80 Hrs	600
<input checked="" type="checkbox"/>	106	SAPComm	120 Hrs	600
<input checked="" type="checkbox"/>	104	RRRS1-Change	120 Hrs	550
<input checked="" type="checkbox"/>	102	RRRS	120 Hrs	500
<input checked="" type="checkbox"/>	101	Chenna	120 Hrs	500

0 ITEMS SELECTED

**UAT / Prod - Backoffice**



## InitialDataSystemSetup.java

In this file, you can specify what ImpEx you want to execute during hAC - Update (or) hAC - INIT?

```
@SystemSetup(type = Type.PROJECT, process = Process.ALL)
```

```
public void createProjectData(final SystemSetupContext context)
```

```
{    ===== What ImpEx Files you need to load ===== }
```

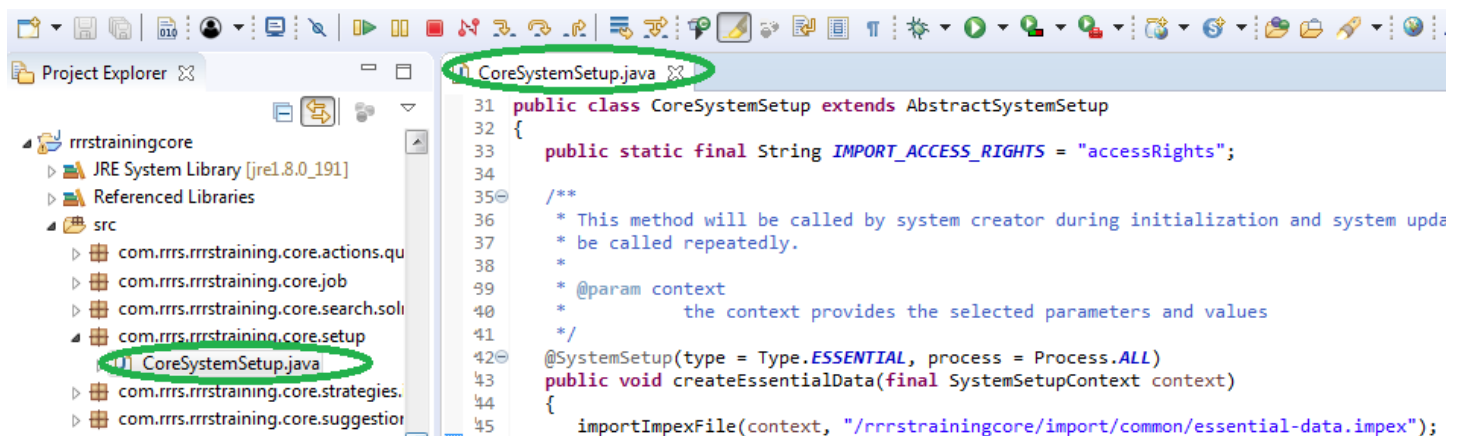
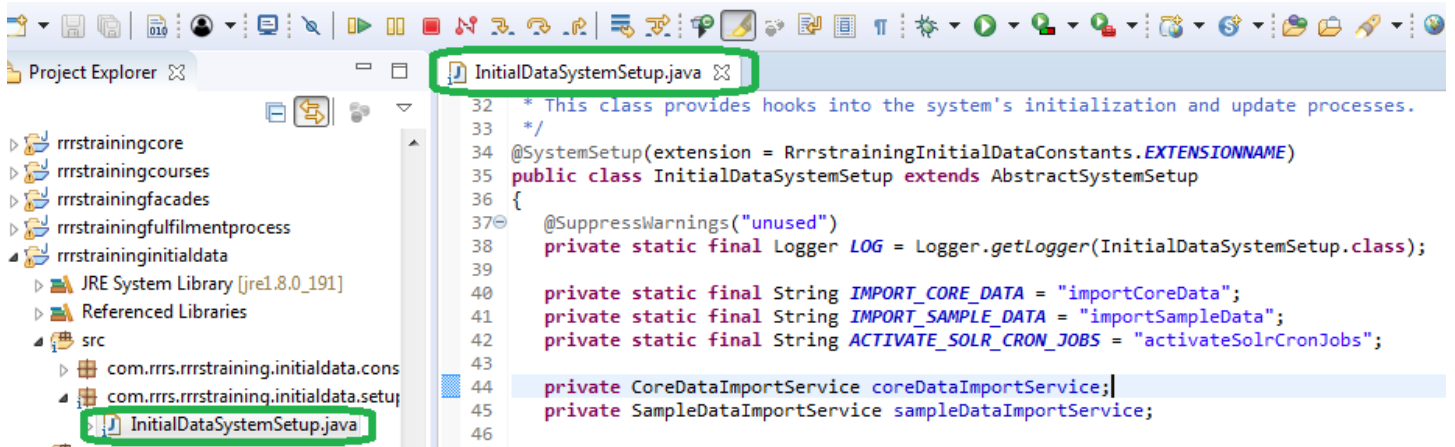
```
@SystemSetup(type = Type.ESSENTIAL, process = Process.ALL)
```

```
public void createEssentialData(final SystemSetupContext context)
```

```
{
```

```
===== What ImpEx Files you need to load =====
```

```
}
```





**Business Scenario** = Generally in companies, each release will have some Impex's. Assume that, we want to allow only that release impex needs to be executed (or) imported. How to provide this flexibility to select the release wise Impex's.

(or) **hAC Customization** Example: -

The image displays a screenshot of the hAC (Hybris Administration Console) interface and the corresponding Java code in the IDE.

**hAC Interface:**

- The browser address bar shows the URL: `https://localhost:9002/platform/update`.
- The interface lists several components for customization, including `rrrstrainingcore`, which is selected.
- Under `rrrstrainingcore`, there is a section titled "Do you want release wise -- RRRS Chenna Custom Data" with a dropdown menu set to `true`.
- Below this, there is a section titled "RRRS Chenna Custom Data to Import =" with a multi-select dropdown menu showing options: `Release1-RRRS`, `Release2-Chenna`, and `Release3-Hybris`.

**Java Code (CoreSystemSetup.java):**

```
134 {
135     return (T) Registry.getApplicationContext().getBean(name);
136 }
137
138 @SystemSetupParameterMethod
139 public List<SystemSetupParameter> getCustomSystemSetupParameters()
140 {
141     final List<SystemSetupParameter> params = new ArrayList<SystemSetupParameter>();
142     //Adding boolean field
143     final SystemSetupParameter customDataParameter = new SystemSetupParameter("rrrsCreateCustomData");
144     customDataParameter.setLabel("Do you want release wise -- RRRS Chenna Custom Data");
145     customDataParameter.addValue("true");
146     customDataParameter.addValue("false", true);
147     params.add(customDataParameter);
148     //Adding multi select drop down box
149     final SystemSetupParameter imports = new SystemSetupParameter("imports");
150     imports.setMultiSelect(true);
151     imports.setLabel("RRRS Chenna Custom Data to Import = ");
152     imports.addValues(new String[]
153     { "Release1-RRRS", "Release2-Chenna", "Release3-Hybris" });
154     params.add(imports);
155
156     return params;
157 }
```



**Q** = If you want to extend / create your own UI component for a cockpit the GUI web framework you have to use is:

- o **ZK**
- o Spring MVC
- o AJAX
- o JSP

**Q** = Each time you implement a custom webservice resource in hybris to complete the build and deploy process you must

- o **Run the ant webservice nature task**

- o **Run the ant clean all task**

- o Edit the relevant spring.xml file and run ant

- o Do nothing. New webservice resources are auto compiled and hot deployed

**Q** = The hybris platform is configured to run against HSQL by default, which of the following is true?

- o HSQL is the recommended DB to use for production environments

- o Configuring HSQL to run in memory is a useful configuration for debugging transactions

- o **HSQL is only recommended for development or test environments**

- o Configuring HSQL is run in memory is the recommended way to guarantee good performance in clustered production environments.

**Q** = When defining enum types for use by hybris item type which of the below is true?

- o Enum types must be defined in java code before being reference by hybris generated item classes
- o Enum types can be defined by using the <enumtype> tag in relevant –items.xml
- o Enum types can be defined inside the <itemtype> tag of the item type which needs to reference them
- o A default value can be set using the <defaultvalue> tag in relevant –tems.xml

**Q** = When your extension has dependency on one of the platform extensions (like validation or catalog) you need to:

- o Use the requires tag in extensioninfo.xml file of your extension
- o Use the requires tag in local.properties file of config project
- o Usually nothing, dependencies on platform are resolved by default
- o Use the import extension in your java code to import the extension you wish to import

**Q** = In which case you must restart the hybris server for your changes to take effect

- o After initializing the system data for first time
- o After creating or updating cockpit configuration file
- o After adding a new spring bean to one of spring configuration file
- o After editing any JSP files

**Q** = In the development environment you wanted to configure a setting to specify a JDBC url the best place to put it would be

- o The extensioninfo.xml file of your extension
- o The project.properties file in platform bin directory
- o The localextensions.xml in configuration directory/Eclipse project
- o The local.properties in configuration directory/Eclipse project

**Q** = Which of the following is true about the configuration setting in project.properties file in the server platform directory?

- o System configuration setting should not be added to this file, because the file will be overridden when you update the platform version
- o After any changes are made to this file a complete clean and rebuild should be performed using ant clean all
- o Any local properties you set in your extensions or in the configuration folder/Eclipse project will override any properties in this file

**Q** = Which of the following are bad principles of OOD?

- o Tight coupling of objects for robust code
- o Tight cohesion of methods for modular approach
- o Encapsulation of data to enable loose coupling
- o Naming classes with verbs or doing words like clean Database or Verify Customer to make it clear to other developers