openSAP
SAP Business Warehouse powered by SAP HANA

WEEK 2, UNIT 1

00:00:13 Hello and welcome to Week 2, Unit 1, Data Provisioning Options for SAP BW.

00:00:18 In this unit we are going to give you an overview of the different source types for SAP BW – so the source system connectivity of SAP BW.

00:00:26 And we'll specifically focus on so-called ODP source systems, which are a new way of connecting source systems to BW.

00:00:36 So, on this slide you see a variety of options that allow you to connect sources to BW.

00:00:44 The first and maybe most prominent one is the connectivity of SAP ERP systems to BW, which is classically based on extractors.

00:00:52 That's right. So, I think that's one of our most or the most famous use case of connectivity between an SAP BW and an SAP ERP system

00:01:05 because the extractor is an encapsulated business object, representing a few source tables already in the ERP source

00:01:15 and what it brings you is, like I said, this business logic view into the SAP BW, and therefore it's really a strength of BW,

00:01:24 which most probably never go away because it's very easy to get the business logic extracted via this structure into the SAP BW.

00:01:34 They also contain or respect all the information which you have made in your customizing, like the fiscal year variant or fiscal variant - all this kind of stuff.

00:01:43 All this configuration has an influence on the extractor, which is very hard to mimic if you only use an extraction based on database tables.

00:01:55 I think in an SAP-driven landscape this is one of the key differentiator between any data warehouse and the SAP BW.

00:02:02 But I think the other ways are also underestimated, because we have many other technologies

00:02:08 like the database connect technology, the DB connect, which allows you to connect to external databases.

00:02:14 It's very generic. So, it's based on SQL, allows you basically create an SQL connection between your BW application server and the external database.

00:02:25 And you can, in a very generic way, extract data from most of the traditional database systems into BW.

00:02:33 File systems are very similar. Also a very generic way of connecting, of loading, say, CSV files into BW.

00:02:41 Flat files, anything which is local and contains any structured data in a way that is necessary to load up into the BW.

00:02:51 Web services are very similar and so it's very fundamental and often used. I think there are many external tools or applications calling the web service interface

00:03:03 because it is an open standard, so it is very easy to call data or to load data into the BW and to get it out from there.

00:03:13 So, this is a very nice open interface and I think the new one, the operational data provisioning connection,

00:03:21 and this is the very new technology here in with SAP BW 7.4.

00:03:28 As you see in the slide. it basically provides some connectivity which existed before, so you can connect ERP systems via ODP to your BW systems,

00:03:37 which is just an alternative way to the existing way. We will later see what the benefits of this new approach are.

00:03:44 Also you can connect BW systems to your BW system using this connectivity and we will see later that this has tremendous benefits.

00:03:53 And, it's a very, very strong framework and therefore we have also another opportunity and this is to connect the SAP Landscape Transformation replication server.

00:04:03 So, this is the triggered based technology. And also via this technology we are able to load data into BW via ODP.

00:04:11 Then BW also provides connectivity to ETL tools like a specific and optimized connectivity to SAP data services.

00:04:21 Right, so this is a standalone ETL tool, also able to load data from a variety of sources into the BW.

00:04:32 So even data services have so many different source connections that with the combination of both

00:04:40 I think we can nearly load any source into the SAP BW; and we have also other two last interfaces.

00:04:50 Here it's an open partner API so where external applications - maybe developed by SAP partners can write their data into the BW.

00:05:00 And there is the SAP HANA Smart Data Access which is again very, very new with 7.4

00:05:07 and - or in combination with SAP HANA; and this is really an interesting technology.

00:05:14 We will also have an own unit for that because it allows us in BW to work in a logical data warehouse fashion.

00:05:22 It's connectivity which also allows you to connect remote databases to your BW systems so it's in that way similar to DB Connect.

00:05:29 But it also allows you to connect big data instances like Hadoop to your BW system.

00:05:35 And even you see here the also streaming options, so we also have tools where you can have a real-time,

00:05:45 a never ending streaming into SAP BW, and this would be the technology Smart Data Access.

00:05:52 This is in general our interfaces and via these interfaces we can do now a direct access of the data, we can load the data in a batch mode in a scheduled way.

00:06:01 This has nothing to do with the technology and therefore we are now focusing only on the latest technology which is then the ODP framework.

00:06:09 Maybe we should mention that ODP and all the other ones except for Smart Data Access are not HANA-specific.

00:06:16 Right. Smart Data Access is based on a HANA technology and therefore only exists in BW on HANA. All the other variants also exist on any DB. Right.

00:06:27 So, what's the point of Operational Data Provisioning or ODP. ODP source systems provide a unified concept, renovating and unifying existing connectivity.

00:06:38 So, we saw this before in the slide that they provide a new way to connect ERP and BW systems to BW and as you're going to see we have a unified way now.

00:06:51 Basically, it behaves the same, whether you connect an ERP or an external BW system to your BW system.

00:06:57 And we also saw already, when you mentioned SLT, that we extend via ODP the range of, the variety of source system options.

00:07:08 From the BW perspective it's a very flexible framework and this framework is really allowing us to unify the data transfer between a source and a target.

00:07:20 So in general, it's based in a certain NetWeaver installation, it's of course depending on the release there.

00:07:27 But then it's really defining the data transfer between a so-called provider. And this can be, you see this in the graphic, different types.

00:07:37 It could be such a trigger-based technology, it could be an ERP extractor, it could even be an SAP HANA view,

00:07:44 means a view what you modeled in a native environment - analytic, attribute, calculation view, it could be a source BW.

00:07:50 And then this ODP technology, that framework has an owned queue, the operational delta queue

00:07:56 which is really an own physical queue in this framework encapsulating the data for a subscriber.

00:08:05 So, the subscriber is, so seen, the target system. It could be data services as ETL tool, it could be a BW system.

00:08:14 And the clue there is really that we can have one provider with many subscribers.

00:08:20 So we can share, so seen, the same queue for delta. And this is really one of the main benefits of this use case here.

00:08:28 Another benefit are the improved monitoring capabilities, so if you are familiar with the classical connectivity of ERP systems to BW,

00:08:35 then you're probably also familiar with the transaction RSA7 in the source system,

00:08:40 which provides you a basic overview of the delta state of each data source there.

00:08:45 In the ODP context we have renovated this and it's now a much more elaborate transaction which allows you to much better monitor the delta state.

00:08:55 Really nice monitor there, with many capabilities, you easily see how many delta subscriptions you have.

00:09:01 And there you can also define, very flexible, recovery and retention periods for each queue.

00:09:08 Means, for each source we are bringing or we are transferring, you can define how long the data is being kept there. And everything else is then done automatically.

00:09:16 So, that's also much more flexible than it used to be. You already mentioned the support for multiple subscribers.

00:09:23 And another point is that in many cases PSA becomes optional on the BW side if you use ODP as a connectivity here.

00:09:32 That's basically because ODQ has a lot of capabilities which we used to have in PSA and therefore now we can basically skip the PSA layer.

00:09:43 You see that here in the overview. So the data transfer process into BW can work or can write directly into an InfoProvider, so seen, without the PSA

00:09:54 because this ODQ, which sits than in every time in the source, provides us also the services of a PSA.

00:10:02 We can recover the data very easily request-wise and we can even do that subscriber by subscriber.

00:10:10 So, it could also be that we have many BW systems, for instance, sitting on that one ODQ.

00:10:15 From a modeling perspective things will not change, so you still model a DTP based on a DataSource into, say, DSO.

00:10:23 But you can basically check a flag, which says Bypass PSE when you load and then data will not be transferred into PSA first via info package

00:10:31 and then transferred via the DTP, but the DTP directly reads from the source.

00:10:37 The picture here shows also the other two use cases. So we use the same idea, the same framework, the same technology also since BW 7.4 for data transfer

00:10:48 between a BW system to another BW system and we can leverage this for the trigger-based technology SLT which also will be shown here in this course in an own unit.

00:11:01 OK, so let's get ready for the demo. Just to give you a quick overview of the demo and what is going to happen later in the system.'

00:11:08 We're going to show you how to create an ODP DataSource based on an extractor.

00:11:13 Then we are going to show you a lean data flow based on an ODP DataSource. Lean basically means it's a dataflow bypassing PSA.

00:11:22 And we will show you the capabilities of the monitoring tool for the operational delta queue.

00:11:30 That should not be much different. So it is really, and that is the good thing, it's a technology

00:11:37 and in the BW world, you see this here, it's really been transparent. So we have different source system, it's called context for the ODP case.

00:11:47 So we have all the context we explained already are reflected here in the system.

00:11:52 And then you can create a source system which is then the connection to the source via this context.

00:11:58 Exactly. So here you see the list of all source system types, which you ought to know from RSA1.

00:12:05 If you haven't seen a BW 7.4 system yet, then the ODP source system types will probably be new to you.

00:12:12 Anyway, we created a connectivity here, an ODP source system connectivity to actually the same system we are using here.

00:12:19 That's basically because, if you want to play around with this yourself in a cloud instance of a BW system,

00:12:27 then you can basically use the same way to play around with this technology.

00:12:33 Then our... This source system should already be there and you can try to load data via it.

00:12:40 Now, let's go into the source system and create an ODP DataSource. So we click on the application component, and we say Create DataSource...

00:12:49 We choose the name of the operational data provider, so that is the object in the source which provides the data.

00:12:54 In this case it's just an extractor, or a DataSource in the source system, right? It would be the same for any other extractor by the logistic cockpit

00:13:02 or whatever... financial, controlling... It's exactly the same, it's just a generic extractor here. And this is the only new step of this whole ODP exercise.

00:13:11 So we are now choosing as ODP the source table and then it's behaving like a source, a DataSource in BW.

00:13:20 So, if you think about the other contexts which you mentioned already, for example about HANA analytic views, then you would provide the name of an analytic view here

00:13:28 which is the object in the source. If you talk about SLT, then it would be the name of the table in the source system.

00:13:36 Then the DataSource needs a name on BW side, and then clicking this one, Transaction Data, so that's all we have to do here.

00:13:44 And then you also see that this is very familiar and looks basically the same as all the stuff before. Right, and if you go to the Extraction tab,

00:13:54 it also shows you the same delta process, methods, means After-Images. And it's really depending on the source.

00:14:02 The ODP, it's really just the way how we transfer data in a very compressed way.

00:14:09 And via this we also should see a performance improvement for existing extractors because the way how we transfer data is now different.

00:14:20 So here in the Adapter line you actually see what type of extraction you're doing here.

00:14:25 But that's it basically, so we can activate this. I get some errors because the internal format is...

00:14:31 So it's checking the source format and the target, but let's load data. Let's load data.

00:14:36 Alright, so obviously we have prepared something for you here. So here we have a dataflow. Let's look at this dataflow in detail.

00:14:43 We have a DataStore object, a transformation, and a DTP. And if we look at the DTP, here is the DataSource.

00:14:52 Now we have to go there. Let's look at the DTP, see what it says. It's a DTP type Delta.

00:15:02 And here, down here, now that's not very easy to read, but it says: Data Extraction: Directly from source system. PSA not used.

00:15:09 This exactly the point where we are now jumping over the PSA table and writing directly into the connected InfoProvider, which is in this case, it's an old DataStore object.

00:15:21 So, many of our demonstrations will be shown on the standard DataStore object,

00:15:27 but, if you heard us carefully in Week 1, there is also another, a new DataStore object type, the advanced DataStore object type.

00:15:36 We will cover this in an own unit, and you will have the chance to see this as well.

00:15:41 So, let's run this extraction, let's push the button and see what happens.

00:15:46 As always you jump into the DTP Monitor. And I guess while we are doing this, we can already...

00:15:53 I mean it's loading now data, which is basically exactly the same technology as it was before. It's the data transfer process.

00:16:04 What is new is really the data monitoring about it. So this delta queue monitor is really interesting. You see here again...

00:16:13 So remember, this is the new version of RSA7. If you compare this to RSA7, you can see a big difference.

00:16:18 It's looking a bit different. You see here again the different sources, the different providers.

00:16:23 We see how many requests we have in there, how many subscribers, means how many targets called the same data.

00:16:31 And if we now go to BW DataSource, you will see our extractor here, which is this one, you see.

00:16:40 And if we are now calculate also the data volume, you will see how many data records have been transferred, how many are in the source.

00:16:47 This is really new. This was not possible in that way before. And if you double-click...

00:16:51 Maybe one thing to look at, you mentioned compression and the way we store data in the queue.

00:16:57 So you see that we have a compression rate of almost 99 percent here

00:17:01 The original size is something like 5, what’s that, billion bites and the compressed size in the ODQ is 77 million.

00:17:13 The very nice thing, if you double-click on it, you will see now the different subscribers. The subscribers are in our case now all the DTPs of our BW.

00:17:23 And here is also our DTP we executed just a second ago.

00:17:29 We can actually see this from the time stamp here in this description. And we see all the different requests.

00:17:36 We can really monitor everything what was happening between the target and the source.

00:17:42 And you see there even that we had one initialization of our data transfer, which brought over 6 million rows, and then we did a delta load and so on.

00:17:55 And this is now, if you click on Refresh and so on and you load data, you will see that every time we execute a DTP,

00:18:03 an info package, and whatever we have new and will be there. So here you, for example, also see that the extraction is still running.

00:18:08 So, this gives you a lot of information about the state of the extraction, all the connected systems, all the connected subscribers, all this information is contained in there.

00:18:17 And this is your administration perspective if you would like to create a data retention period and stuff like that.

00:18:31 So, what are the key takeaways of this unit? You have seen all the source system types, which exist for BW.

00:18:37 You have seen the new source system type, called ODP, in all its variants. And you have probably seen some of the main benefits right away.

00:18:47 Of course, if you play around with this on your own, you will see more benefits.

00:18:51 You have seen that there is an optimized connectivity now between ERP and BW systems.

00:18:56 And you've seen that we also extended the range of connectivity for BW using ODP technology.

00:19:03 Without that it's looking much different, it’s still the same, so ... I think this was this unit, so have fun with your self-test for today.

00:19:13 And see you in the next unit, where we will talk about other ways, alternatives for data provisioning which will be interesting.

WEEK 2, UNIT 2

00:00:13 Hello and welcome to week 2, unit 2: Integration of External Data Structures with Open ODS Views.

00:00:19 In this unit, we are going to introduce a new BW object type, which is called Open ODS views

00:00:25 and we'll show you how you can use this new object type to consume external data in BW and also to combine this external data with data in your BW system.

00:00:34 I think that's the key aspect here. So far, we were always loading the data into BW, means the ETL process,

00:00:44 and then persisting it in BW as an own data persistency and object.

00:00:49 With this new object, which is totally BW 7.4 and SAP HANA-related,

00:00:57 we have an option for directly consuming data sources, data structures, of external sources directly with giving them special semantics.

00:01:07 This is really interesting and new to see how this is going on.

00:01:11 So let's dive into details a little bit. What are Open ODS views?

00:01:16 Open ODS views are virtual objects that allow you to define analytic semantics without using InfoObjects.

00:01:23 We'll come to the details of this later and what this really means.

00:01:27 They also allow you to use analytic functionality on top of external data structures in a very simple way.

00:01:33 Analytic functionality is all the stuff you do using the OLAP engine, the OLAP processor. So all the calculations and all the functionalities which you use in the Query Designer,

00:01:42 all of this can easily be put on top of external data once you use Open ODS views.

00:01:48 This means we have here really an integration possibility of external sources, external data into BW

00:01:54 by using all the BW services we introduced to you already. And we will give you, of course, more details, like on the OLAP engine.

00:02:03 But this is really the first time that we are able to work in a field-based fashion. This means we are not forced to create an InfoObject for every

00:02:12 column of a source, so we can really freely design and define data definitions or,

00:02:19 in this case, the system is already proposing a definition based on the source table definition and we can work with that in BW as the data would be

00:02:29 in a Cube or in a DataStore object or wherever, but it's a virtual object directly accessing that

00:02:35 and with that, supporting agile modeling approaches and prototyping very easily.

00:02:42 You already mentioned that this allows us to put BW services on top of external data.

00:02:47 Typical services which you might want to use on top of external data are, for example, authorizations. So if you want to run a BW query on top of external data, 00:02:55 authorizations are always a very important point.

00:02:58 You have to secure data. Another point, that's maybe more technical, you might want to have a quick look at data to see if it makes sense to integrate this into your BW data warehouse.

00:03:08 And you might actually during this process detect that data types don't really match or the formats don't really match.

00:03:15 You might find out that it really is necessary to transform data and create the persistency.

00:03:21 With Open ODS views, this is also supported. So you can start working with data in a very simple and lightweight fashion,

00:03:27 and once you see that you have to tweak the data and make it conform with the existing data in your data warehouse,

00:03:33 Open ODS views also provide you with this.

00:03:35 I think a classical example is I do have transaction data in several sources. I have my SAP BW as system with governed master data, harmonized and so on.

00:03:48 and I would now like to see if my external source with transaction data is fitting to my harmonized master data because there my master data

00:03:57 is really the corporate view and via these features, I can do this in less than a minute. We will prove that in a demonstration later.

00:04:06 Now let's come to the classical BW and quickly review the classical BW modeling paradigm

00:04:12 which is based on InfoObjects. So when you create any object in BW, you will have to start with InfoObjects.

00:04:18 For example, a simple field is always represented by an InfoObject. An InfoObject basically defines the data type of this object. It also defines

00:04:28 the information like is it a characteristic, is it a key figure, which is very important for analytic purposes because characteristics are used for drilldown, key figures are used for aggregation.

00:04:39 So this distinction is directly made in the definition of an InfoObject.

00:04:43 Other properties of InfoObjects, especially in the case of master data, are important. Like text information. You might have language-dependent text.

00:04:51 Hierarchies, all this kind of stuff. All this is encoded in an InfoObject

00:04:55 and you basically load data from sources into either InfoObjects or InfoProviders which you again construct out of InfoObjects.

00:05:04 Right. I think that's a good thing, actually. So don't get us wrong. InfoObjects are actually a really good thing.

00:05:13 But in case I would really like to, in an agile way, bring up a new data set, it would also mean that if I have a table with 200 columns,

00:05:23 I have to create 200 InfoObjects. And this was complained by many customers, by many users, that this is not agile, not fast enough.

00:05:32 We can now do this the way around if you go to the next approach, if you focus on how we do it with Open ODS views.

00:05:40 Open ODS views actually take the other way around. You are given a source structure or multiple source structures

00:05:45 which come with field names and data types. And you only add the information which is missing, the analytic semantics, as we say.

00:05:53 So for example, you start with a table, look at it, and you have to decide whether it's master data or transaction data, whether it's facts or dimensions. That's part of semantics.

00:06:01 Or if you go to the field level, you look at a field and decide this is more of a characteristic or it's more of a key figure.

00:06:09 So you use the metadata which comes from the source and you only add what's necessary.

00:06:14 That's the idea of this modeling approach. Thereby, it's basically...

00:06:18 Yeah. But I think that means also that if my source—or my data model in the source—is already representing a data model like we have it in our EPM scenario

00:06:28 where we have a fact table, the sales order table, and I have related master data like the business partner, like a product, and so on.

00:06:36 I would like to not only virtually access the sales order because those are just my facts but I would also like to see what is the name behind the customer ID or something like that.

00:06:45 And I can, via the semantics, we can associate the different tables to each other and this is exactly what the idea is about of connecting the Open ODS views to each other.

00:06:55 That's what the green lines here try to indicate. You might think of one Open ODS view representing the sales order facts

00:07:04 and one Open ODS view representing, say, the business partner master data. Of course, there is a certain link between this.

00:07:11 There is one view called Business Partner in the facts and you have additional information about this in the other table.

00:07:18 And linking this by an association basically allows you to leverage all the attributes of the master data field in all the reporting you do.

00:07:29 When we speak about source, it means that this Open ODS view can consume the data out of many different sources.

00:07:37 And we have different technologies for accessing the sources.

00:07:41 It could be, for example, the same HANA database, so meaning we are accessing a second HANA schema next to the BW where the data is located.

00:07:50 It could be a better source in BW so that we can access whatever, an ERP system or something like that.

00:07:56 Or it could be related to the SAP HANA smart data access, which is a brand new technology where we will focus in the next unit.

00:08:06 But this is allowing us to directly consume, for instance, sources like a Teradata instance, like Hadoop sources, and stuff like that.

00:08:15 I guess we are ready to jump into the demo so that you really get a feeling of what this looks like.

00:08:20 The demo will, by the way, be in the Eclipse environment. So Open ODS views can be modeled inside Eclipse. We will see this in a second.

00:08:27 We'll show you, first of all, an existing Open ODS view and explain the details of what it really means. Then we are going to create an Open ODS view as well.

00:08:37 We are going to show you how you can combine Open ODS views with existing master data InfoObjects.

00:08:41 I think in this case it will be the product. We will show you how to use facts from an Open ODS view and combine them with master data from the product InfoObject.

00:08:50 Then I guess it's time to jump into the system.

00:08:54 And I think it's important to mention that—I think you hopefully learned already in the course—that this new Eclipse-based modeling environment is totally SAP HANA-related.

00:09:05 This means the Open ODS view is a HANA-related feature because we need certain functionalities of the database. We need a certain speed

00:09:13 for doing the temporary calculations in the background on the fly. That's really the reason why this is exclusively with SAP HANA.

00:09:21 And the features we are showing are also related to the latest BW release: SAP BW 7.4 SP8.

00:09:29 Okay. So let's have a look at the Eclipse environment, here the BW Modeling perspective in the Eclipse environment.

00:09:35 When I open an InfoArea here, and here in my Favorites I have a number of InfoAreas,

00:09:41 we're looking at the sales order Open ODS view. If you look at the Overview tab of this Open ODS view you see some general information.

00:09:51 The most important information is that we have assigned the semantics facts to this because it's transaction data, so this has the semantics facts in a star schema. This would represent the center of a star.

00:10:02 That also means if we later would do reporting in top of this object, we would have the functionality of a key figure available

00:10:10 because the key figure is something we think, hey, this is fact-related to a fact table, so we have certain functionalities here now available

00:10:18 in comparison if we would choose the master data type, which will be our next example.

00:10:24 So here on the left-hand side you see the structure of the source object. The source object in this case, if you look at the top line here, says it's a database table or view.

00:10:32 And here you see the name of the view and the database schema in which it resides.

00:10:37 Here you see the names of the fields and you actually also, if you scroll a little bit, see more of the technical information about the field, namely the data types and length.

00:10:46 From this information, the system proposes certain characteristics or certain properties of the fields.

00:10:54 For example, by analyzing the data types, we make a proposal whether a field is rather a characteristic or a key figure.

00:11:02 That's not always correct because, for example, integers could be potentially both.

00:11:06 You could possibly want to drill down by an integer or you could want to use it as a counter, for example, as a key figure.

00:11:13 But the system does a proposal. And then, on the right-hand side, you can basically...first of all, you can change this proposal by dragging and dropping fields between these different folders.

00:11:24 And on the right-hand side, you can basically fine-tune the information about this field.

00:11:29 So you could also override the name.

00:11:31 You can change the name, you could change the description.

00:11:34 You can, for example, define whether a field is authorization-relevant. All this is done here.

00:11:38 This is then going along with the BW authorization. This is again an integration point between the BW authorization feature and this directly consuming Open ODS view.

00:11:49 All right. So the first thing we actually wanted to show you is how to bring this transaction data from an Open ODS view together with the product master data which we have in an InfoObject.

00:11:59 So what you do here is you select the PRODUCT_ID field and you associate it with an InfoObject.

00:12:08 I think it's one of the content InfoObjects which is delivered with the Enterprise Procurement Model. It's called OD_NW. Let's see.

00:12:18 Why are we doing this? Because we have already a harmonized master data object here in BW where we are sure that we have

00:12:28 cleaned and harmonized master data and we would, of course, like to leverage this to see

00:12:34 and to assign now the IDs popping up out of this Open ODS view with our master data in BW. And we can reuse all the services like, for instance, the navigation attributes like authorizations, and so on.

00:12:46 And maybe the most fundamental one should be here that we expect to see a text now because this product InfoObject certainly contains a language-dependent text in the text table which we loaded before.

00:12:57 So if we activate this and start a query or go into Analysis for Microsoft Office,

00:13:05 we should probably, if we drill down by product ID, see the description of the product and not the product ID which is actually contained here in the transaction data.

00:13:15 So let's see what happens if we open this Open ODS view.

00:13:19 How long is it now taking?

00:13:22 Let's say two minutes, three minutes until we are able to query the first time this object in BW.

00:13:31 So let's see what happens if we open this Open ODS view. It's represented as an InfoProvider under the same name.

00:13:38 That's the name of the Open ODS view and that's also the name under which you see it in, say, the Query Designer or Analysis for Office.

00:13:47 Let's open this. The system now runs the default query on this InfoProvider which uses all the

00:13:55 key figures in the columns and gives you all the characteristics as a potential drilldown.

00:14:03 Now let's see what happens if we drill down by product ID. Where is the product ID?

00:14:10 Here it is. Let's see what happens if we drill down by product ID. And as I said before, we would expect not to see the key, but actually the

00:04:20 the text representation of the InfoObject.

00:14:23 Again, this is really done on the fly. So we are accessing the external source and on the fly

00:14:29 we leveraged also a bit of the OLAP capabilities by assigning the texts to the ID.

00:14:36 And here you go. You see the text assigned instead of consuming the raw data directly.

00:14:42 All right. So that was the first step in showing you how to combine external data of Open ODS views with BW InfoObjects.

00:14:51 Now we could also think of another approach, for example if we take the business partner here.

00:14:58 The business partner in the EPM model is of course also represented in a table, a database table representing the business partner master data.

00:15:05 So why not use an Open ODS view on this database table and associate this Open ODS view to the field Business Partner here.

00:15:14 Just as we did with the InfoObject, we could also, as we saw, associate another Open ODS view.

00:15:20 From a functional point of view, this should actually be the same.

00:15:23 So let's see how we can do this.

00:15:28 So what we are doing is we create a new Open ODS view.

00:15:32 And the idea is really that we link this new Open ODS view, which will represent the master data,

00:15:38 together with the facts represented by the other Open ODS view because the source is already having this association and we are representing it now

00:15:46 in BW and have a model built there via two Open ODS views on the fly.

00:15:50 Right. So here, we first associate certain multidimensional semantics to this object. We said well, the business partner is probably

00:15:59 of type master data. It's a dimension object.

00:16:02 So let's do this. Then we select the source. So we have to find the table and define it as the source of this Open ODS view.

00:16:10 So we pick Database Table or View. Now unfortunately,

00:16:17 we are very close to our development system so we have a small bug here. You see that it states DataSource and source system

00:16:23 where we actually expect something like database object and database schema.

00:16:28 What we have to do is pick this here even though we're not using HANA smart data access in this scenario. We're really using a database object.

00:16:37 But be patient. This will be in the next unit. We do exactly this.

00:16:42 It's going to be funny to see which option we have to get the HANA smart data access then.

00:16:47 So what's the name of this table? I kind of memorized this. YSHEPM. Not really memorizing.

00:16:57 Let's see. Here the system proposes all the tables or views with this prefix. BPA is certainly the one which you're looking for.

00:17:07 And there you go.

00:17:10 Again, here you see that now we have assigned the semantics master data to the whole object.

00:17:17 Let's increase the size here. On the left-hand side it's again the same. We have different view names, of course, because it's a different structure.

00:17:24 But in principle, you again see the data types and length information. And the system has made a proposal,

00:17:29 assigned some of the fields to Characteristics, some to Key Figures, and actually detected a Currency.

00:17:36 because the table apparently...well, it's a table from the Data Dictionary, so it provides more information than a plain database table would provide.

00:17:44 So we know already that there is a currency code in there and we should not stupidly aggregate all the key figures because it doesn't make sense. And we would expect,

00:17:52 like in a query, what we also demonstrate later, that it would be then assigning a star for hey, I cannot aggregate the data.

00:18:00 Yeah. Right.

00:18:02 So now, especially in the case of master data, it's very important to maintain the keys correctly.

00:18:07 We have some folders for keys up here. So there's the BP_ROLE, which is a key field.

00:18:12 That's something you have to know, you have to study the EPM model. That's something we did before, obviously, and we are familiar with it.

00:18:20 And what's also nice here is we could—

00:18:24 also for demo purposes—maybe look if one of the fields here...if it makes sense to use one of the fields on the left-hand side as short, medium, or long text.

00:18:35 And if we look at the fields here, maybe COMPANY_NAME would be nice. I mean, if you drill down by customer or business partner, then seeing the company name of the business partner is probably a good idea.

00:18:44 This means also that that can assign one field of the source multiple times through the target structure.

00:18:50 And it could even be that we have a field representing a key figure and a characteristic at the same time. This is also totally new for us in the BW world.

00:19:00 All right. I think that's all we have to do here. So we activate this Open ODS view.

00:19:04 And now we're going to go back to facts, to the previous one which we looked at earlier.

00:19:11 And we're going again to the Business Partner field and now associate the Open ODS view which we just created.

00:19:17 So the name was...you can still read it here. YEPM.

00:19:20 This is now establishing the connection between the two views.

00:19:25 If we drill down in the view, you will see that it will jump into the definitions and the characteristics maintained in the other view.

00:19:34 Okay, now we have to maintain the join condition correctly. We have two key fields and we have to make sure the join condition is maintained properly.

00:19:43 That's a pretty easy thing.

00:19:45 And now let's see if we can use one of the fields of the business partner structure as a navigation attribute, for example.

00:19:54 So we used the COMPANY_NAME already for other purposes. Maybe let's take the LEGAL_FORM here.

00:20:01 Legal form. You would then expect, if we run the default query again, to see this field in the drilldown options.

00:20:11 So. Legal form here.

00:20:13 We activate this.

00:20:16 And again, that's all we have to do. Now we are basically ready to jump into Analysis Office.

00:20:23 And we reload the query and we will see if we see the text and stuff we assigned there.

00:20:30 Open the DataSource, so open the Open ODS view. That's the fact Open ODS view and the fact that we added this association should now be somewhere reflected in certain places

00:20:40 in this drilldown. So what we see here is we see the legal form as an additional drilldown possibility

00:20:48 because it's actually joined from the master data to transaction data.

00:20:53 And we should also see that the business partner is now available with a different name. If we drill down by business partner, we should not see the business partner ID,

00:21:02 which is probably an integer or something, I don't know. But we should see the text.

00:21:07 And the text should actually be the company name.

00:21:13 So you see that adding associations and assigning semantics like facts and master data to the objects themselves,

00:21:22 that's basically enriching the given structures to really make it more similar...

00:21:29 enrich it with information which you already have normally in a BW model so that the analytic engine of BW can easily consume it.

00:21:35 And, as we expect here, we see the names of our business partners.

00:21:43 One aspect which I also would like to mention here—maybe let's remove this drilldown for the moment—

00:21:48 Maybe you're wondering why you are seeing the stars here. That's also something which has to do with the semantics we added.

00:21:56 If you look at the key figures here, for example, the GROSS_AMOUNT. The GROSS_AMOUNT has a currency code associated, that's this CURRENCY_CODE field. You mentioned this earlier.

00:22:07 If we look at this here and drill down by currency code,

00:22:16 then we should actually see the star replaced by the individual currency codes and the summary line will still contain the star

00:22:25 because the system knows that it doesn't make sense—here you see the currency codes—and the system knows that it doesn't make sense to sum up these individual numbers

00:22:34 because they have different currencies associated.

00:22:36 And this is now pure BW functionality based on an external source.

00:22:40 Right. So I guess that's it for the demo.

00:22:45 So what are the key takeaways of this lecture?

00:22:49 Open ODS views provide an easy way to consume and integrate external data with BW data

00:22:56 without using any InfoObjects. But you can leverage existing InfoObjects, so it's a very quick way to connect BW to external data

00:23:04 and it still allows you to grow into your existing BW infrastructure.

00:23:10 In the next unit we will see how we can extend this further by a new connection type. This is the next unit.

WEEK 2, UNIT 3

00:00:13 Hello and welcome to Week 2, Unit 3, SAP HANA Smart Data Access for SAP BW - Logical Data Warehousing.

00:00:20 In this unit we are going to address some of the key aspects of the concept of logical data warehousing.

00:00:27 We will show you the technology on HANA side which supports this data warehousing methodology or paradigm.

00:00:33 And we will show you how BW provides functionality on top of the Smart Data Access functionality of HANA

00:00:40 in order to give you a kind of streamlined and nice experience from a data warehousing perspective.

00:00:48 So what's the point about logical data warehousing, where does the concept come from and what are the problems it tries to solve.

00:00:55 Maybe one introductory point here. The term logical data warehousing was introduced by Gartner about 2 or 3 years ago.

00:01:03 And if you do research on the web, you'll find a lot of useful information. We can only touch the key points about the topic here.

00:01:10 We only have a few minutes for this but we'll try to give you a feeling of what it's about and why it's very useful.

00:01:17 I think in general, the concept comes from a few frequent complaints in the past

00:01:24 that a data warehouse should be able in an agile way to integrate new data sets or data from external sources.

00:01:34 A good example is, for instance, you have a data warehouse with harmonized master data with maybe also transaction data which is there.

00:01:44 But there is also other sources which contain transaction data like another data warehouse, a data mart or something like that.

00:01:51 And then a logical data warehouse should be able to federate and to bring these remote sources into the data warehouse as a new source in an agile way,

00:02:02 that a common view for reporting is possible. And of course the point is, if a new source arises in your system landscape,

00:02:15 then the traditional way would be to create a persistency in your data warehouse, start harmonizing the data,

00:02:21 start loading the data so it's a lot of modeling, it's a lot of ETL which is involved.

00:02:26 Well, you already have the data somewhere in place, so using this data in place just to investigate what the effort will be to harmonize the data.

00:02:35 Or if there... Maybe there is no big effort, maybe it already fits, right?

00:02:39 So, having a quick way to decide what the effort is or maybe a quick way of leveraging data if it's easy to leverage, that's the whole point about logical data warehousing.

00:02:49 Now, one thing which you should also keep in mind is that when we talk about federation here:

00:02:56 Federation is very useful when it comes to this kind of agility because it removes a lot of burden from you when it comes to modelling on the data warehousing side

00:03:06 but of course, the point is that you don't have the same service levels when we talk about federated data as with the data in your local data warehouse persistencies.

00:03:16 For example, for data quality, for the time, performance during reporting because the federation aspect is really, it's a virtual access during runtime

00:03:28 but logically, the model should be looking from the BW perspective transparent as one logical model.

00:03:37 And the objects we are using there in the SAP BW world are the open ODS view, which we saw last time. We just learnt a unit before.

00:03:47 And this object is able to call a technology down in the database and SAP HANA is providing us there the Smart Data Access layer

00:03:57 which is exactly the technology for federate virtual reading accesses.

00:04:03 So we have a technology there to provide access to other relational database management systems or even for a big data Hadoop clusters.

00:04:13 And this is really then the connection, the interface for us to enable logical data warehousing.

00:04:20 Maybe one more aspect here: So access to federated data is fine as we just said.

00:04:27 You might find out that it doesn't fulfil all the service levels which you need to fulfil.

00:04:31 Maybe data changes are something which you want to freeze for a certain time. So, that's something you can't do on a federated source which you don't own.

00:04:40 Now the point is that starting with federated data is probably the cheapest and easiest way to leverage these data and to get hands on these data

00:04:48 but at some point, even after you provided certain solutions to your end-users, you might find out that not all service levels can be guaranteed with federation.

00:04:56 And then one of the key aspects of logical data warehousing is that it should enable you,

00:05:02 and that's something where BW provides quite a few services in this area,

00:05:06 the system should enable you to switch from a federated access to access of local data

00:05:16 so basically move the data from the federated database to your local persistency

00:05:21 without a huge impact on older, say, queries, older higher layer object which you have created already.

00:05:27 So basically, this move from federation to local persistency should be fully transparent from a query perspective.

00:05:34 And we should also keep in mind that a federated access is really the virtual access, is the access on the fly

00:05:41 and you have to find really the balance there in terms of performance - what does now make sense.

00:05:46 It is really necessary that there are accesses on the fly every time or is maybe even a persistency faster.

00:05:53 Even if we have a fast technology like the Smart Data Access natively implemented in SAP HANA, there will be the point in time where you have to decide,

00:06:02 okay a virtual access is fine also in terms of performance because you are delegating the question of performance to the remote database.

00:06:10 And if the remote database is not answering the query fast, then you will wait in the BW application.

00:06:17 Right, so maybe let's come to the Smart Data Access technology a little bit. What is it?

00:06:23 Well, I think you mentioned it already, it's a federation technology included in SAP HANA.

00:06:29 It provides access to various sources. Typical sources are other RDBMS systems but it also includes big data installations and big data clusters like Hadoop.

00:06:40 So what are we doing in a Smart Data Access layer? We are really delegating an SQL access to the remote source

00:06:48 and then fetching the result and working temporary on the fly in this database layer via an so-called virtual table in HANA.

00:06:56 This is happening in the background, so nothing you have to take care about. So SAP HANA is managing that for you.

00:07:02 We just have, from BW perspective, to create one time the connection to this remote source.

00:07:08 This has to be also there in SAP HANA and then, from the BW perspective, we can reuse that for direct access scenarios.

00:07:16 We have even a new source system type for creating exactly this Smart Data Access layer.

00:07:25 So, it's demo time again. We will show you the configuration on HANA side and give you a quick idea what you have to do there.

00:07:33 We will also show you how this configuration on HANA side plays together with the source system or the source system on BW side

00:07:40 and why this kind of pair is necessary and what it's good for.

00:07:45 And then we'll show you access to federated data. We will show you how to combine or how we can combine federated data with local master data in BW, for example.

00:07:54 The same would also work for transaction data, obviously, and we will show you the step which I mentioned earlier

00:07:59 that, in case you see that you cannot fulfil your service levels with federated access, how the system supports you in basically creating a data flow

00:08:09 which moves the data to a local persistency and ensures that all the higher layers will still continue to run.

00:08:15 Right, and this is closing the gap of the feature with the open ODS view where we told you already, hey, this is the feature to access the remote instances.

00:08:25 And this is now exactly the technology behind. Right. So let's come to the HANA side of the story, the pure HANA side of the story.

00:08:34 If you go to the HANA Studio and the Administration Console here and look at the HANA system, you remember that there are a couple of folders for Catalog, Content.

00:08:42 And there's a Provisioning folder. If you open the Provisioning folder, this contains so called Remote Sources.

00:08:49 And in the Remote Sources, exactly what we are talking about: When we're using Smart Data Access,

00:08:55 a remote source is a database connection from a local HANA instance to a remote database or maybe a big data cluster like Hadoop.

00:09:03 I mean, what we are doing there is really just providing the connection types. So the adapter for the connection is there which is depending on the database we are accessing of course.

00:09:14 We have to provide and save there the access information, means the server parameters, the port, and so on, a user.

00:09:22 And once we have created this remote source on database level, we can then reuse it in the BW perspective as an own source system.

00:09:31 And then for us in BW it's really transparent. We don't have to bother about what is happening behind the scenes

00:09:38 because it's looking here for us like any remote source system with database connect, and the delegation of the reading access is then happening on database level.

00:09:48 So we have prepared such a source system already. Let's have look at the connection parameters of the source system.

00:09:56 If you look at this here, it says Connection Type is HANA Smart Data Access and it actually contains the name of the Remote Source which we just mentioned.

00:10:03 So this source system actually points to a remote database accessed via the remote source HDB_SDA.

00:10:14 And in this database, it points to a specific schema called SAPHBD.

00:10:20 And the point about the source system on BW side, maybe that's worth mentioning, is: Why do we need a source system here?

00:10:27 And why does the, that's something you will see in a second when we create an open ODS view, in such a remote scenario we will have to provide a source system.

00:10:36 The point is, that if you do a transport from your development environment to your productive environment, then probably this configuration will change.

00:10:45 The remote source will be different, the schema might be different.

00:10:49 And to decouple this part of the configuration from the actual open ODS view, we have introduced the source system here.

00:10:59 So it's just also for reusability terms and that we are free also working the transport and all the changes when you copy a system, this is actually where we use this.

00:11:10 And this source system, and this is now also what you've learnt previously with the open ODS view, this is now the connection, the interface

00:11:19 where we have to create an open ODS view based on the source system type and then we are ready to go to consume the remote source directly.

00:11:28 So let's see how we can create an open ODS view on such a remote source.

00:11:33 Let's give it the name YEPM, open ODS view, sales order 3, for example.

00:11:40 So what we are doing here is we are accessing now the again the classical sales order table we are using all the time here.

00:11:48 And now we are doing this with Smart Data Access to make our life easier. You can see and that you, of course, we can also replay this scenario in the cloud instance.

00:12:00 We will federate so seen from the same HANA into the SAP BW system.

00:12:06 So, it's a closed scenario and of course we still have the same small system bug we had already in the previous unit.

00:12:13 Means, we are selecting here of course Smart Data Access but to get the right sub-screen here and the right information down here, unfortunately, we have to switch to Transformation.

00:12:24 If you remember the last unit, to access a local database table or schema, we had to choose HANA Smart Data Access as a source type.

00:12:33 Now it's Transformation instead of HANA Smart Data Access. Sorry about that.

00:12:37 So here, if we enter the... If we now have to specify the source object, we can either specify the remote source, the remote database, and the schema, and the object name.

00:12:49 Or we can provide a source system because the source system, as you remember that's what we just showed you in the SAP GUI,

00:12:57 actually encapsulates exactly this. So it's... Both ways are possible. You can specify these three parameters or you can specify the source system.

00:13:06 So what's the name of the source system? If we type now the name of the source system in there, you would see that we call all the other...

00:13:13 There you go. That we have all the other parameters already there.

00:13:16 So again, it's, for us in BW, it's really easy because these parameters have been maintained on HANA side already and we can reuse that.

00:13:25 Now we specify the table which we want to access. That's again, no it's YSHEPM. That's the classical sales order table like I mentioned.

00:13:36 Should be right now. Here's the table. There you go. Okay.

00:13:41 So, that's what you would do here. We've prepared an example and enriched it already with local master data, and we're not going to show that right now.

00:13:51 So, let's have a look at the completed example here. So that's exactly the same thing. If you look at the details here and the source object,

00:14:03 you see it's the same remote source, it's the same remote database, same schema, and the same DB object name.

00:14:10 So that's really the example which we just were about to create, and we've done a little more work here.

00:14:16 It basically feels exactly the same as we were accessing a local database table because all the remoteness is basically encapsulated here in the source object.

00:14:25 But other than that, everything is the same and feels exactly the same, you can do exactly the same stuff here.

00:14:30 And also the source definition, it's now just getting called from the remote source but here from the BW perspective it's really transparent for us.

00:14:38 So, you still have the technical information about the fields which we saw last time. It's still provided here so everything feels exactly the same,

00:14:48 with enriched the product ID here with an association to a local InfoObject: 0D_NW PRODUCT_ID .

00:14:57 So that's the one which provides or has the product master data.

00:15:02 And I guess we are ready to call a query now. So the query will read federated transaction data and combine it with local master data.

00:15:11 Basically, the text will come from the local master data, I think, and all the transaction data will come from the federated access.

00:15:20 And if we now execute the query and take a look on the, first of all, on the performance, it was quite nice because of our remote sources of course also on SAP HANA in our case.

00:15:29 And how is the query result now looking? It's looking a bit boring because actually for us during reporting it's now really a logical unit.

00:15:39 And that's exactly the point of logical data warehousing so far that we are not aware of that maybe even the result is now been fetched from a remote source.

00:15:48 In the query result you see that as one common view coming and provided by the BW, and that's exactly the point what we would like to achieve.

00:15:57 And you see, it's now also been associated with the master data of the product ID and the sales data coming from the remote source.

00:16:05 You see the texts here which certainly come from the local master data, from the InfoObject.

00:16:11 Now, the transaction data, the facts here, the key figures, this of course comes from the federated access to transaction data.

00:16:19 And now we are coming to the point that, ok, this is nice for direct access.

00:16:23 But what is if I would like to change now, for example, the time stamp information and make it to a date information?

00:16:32 Or I would like to change the key to a text, and this is where a persistency comes into it again.

00:16:37 Or maybe the source is just not reliable, the source connection is not reliable because it's really remote.

00:16:42 It's maybe on a different continent, in a different time zone, whatever.

00:16:47 So, what can you do here? Well, for that, maybe let's close this here.

00:16:54 So we can switch the persistency like we said from a direct access to persistency.

00:17:00 For that we have to jump into the ABAP-based version of the open ODS view so this is something what we not mentioned so far.

00:17:08 So far we were just looking via the Eclipsed-based modeling on the open ODS view

00:17:12 but this is one object where you also can do the modeling in the ABAP side but we definitely recommend to do it on the Eclipse side.

00:17:19 But this switch for the persistency is been implemented here in the ABAP.

00:17:24 It will certainly follow in the Eclipse version at some point in time but especially in the open ODS view area the SAP GUI is sometimes a little bit ahead

00:17:33 because that's also for us some kind of prototyping where we can start with an implementation in SAP GUI.

00:17:39 And then, once we're sure how it works and how it should work, bring it to the Eclipse world.

00:17:44 By the way, it feels, it looks very much the same, right, you have the same stream layout roughly in the SAP GUI as you have it in the Eclipse.

00:17:52 So what you can do here is the following. When you go to the Change mode of the open ODS view, a button shows up which says Generate Data Flow.

00:18:01 And this button, let's see what happens, it proposes to replace the source object by a data source,

00:18:12 which basically means that, instead of accessing the remote table directly,

00:18:17 it will logically place a data source between the source object and the open ODS view.

00:18:25 Now, from an access perspective, this doesn't make any difference. The data source is just a logical object so you should not think of accessing PSA or something.

00:18:35 This works similar to direct access of a data source.

00:18:39 In fact the SQL commands or the commands which we issued to HANA at query runtime will stay exactly the same.

00:18:49 But now we are moving to an adapter on which you can start building a BW data flow. That's the whole point.

00:18:56 So if you do this, maybe let's just do it, and you will see how the open ODS view changes if you look at this source object here. So some things get created.

00:19:07 You see that the source system type has changed. It's now a BW data source. The data source is assigned a name, it's still in the same source system.

00:19:15 That's also one of the reasons why we need a source system here when we create a data source. A data source needs a source system.

00:19:21 The query would give you exactly the same result with exactly the same performance because the query access would be exactly the same.

00:19:28 And now, since you have a data source, you have basically the entry point to what's data warehousing in BW.

00:19:34 You can now create a transformation, you can create a persistency, or even better, if you click on the button again, the system proposes to create a persistency for you.

00:19:45 So it would create an advanced DataStore object, which we are going to see in another unit of this.

00:19:52 You have to assign a name for this DataStore object, and it would create a complete data flow based on this data source

00:19:59 with a mapping proposal for the transformation which will, of course, at first be one-to-one but it's a proposal which you can afterwards modify.

00:20:08 You can modify the DataStore object if it doesn't fulfil your needs. For example if you have this issue with timestamps in the source and you want to have date fields in the target,

00:20:16 you can modify the DataStore object afterwards and you can adjust the transformation and everything will work.

00:20:21 And this is kind of also an evolutionary approach to start virtual, to start virtually accessing and consuming a remote source,

00:20:32 but then, time by time, you can then switch to the persistency and then you have really shown a mature dataflow there.

00:20:41 And this is exactly what we would like to have: Start with a logical data warehousing

00:20:46 and end up in the persistency if the data volume rises, if the performance is questionable and so on.

00:20:52 One thing which we should mention: The query we just used would still work with this changed persistency, would still work without any changes

00:21:01 because the query actually resides on the open ODS view, and the open ODS view has not changed in its structure.

00:21:07 Only the source object of the open ODS view would change, and that's fully transparent to the query. So that's the crucial point from my perspective:

00:21:15 That this switch of federated access to a local persistency is fully transparent to all BI applications which you have created on top of this this data set.

00:21:28 So let's wrap up what we've seen in this unit.

00:21:33 BW uses HANA's federation technology which is called HANA Smart Data Access.

00:21:39 And basically enriches it in order to give you a kind of smooth picture and a smooth experience of logical data warehousing.

00:21:49 And I think now it's time for the self-test.

00:21:53 In the next unit, we will then load or we start loading data in a persistent way into BW and we'll do that via a real-time replication.

WEEK 2, UNIT 4

00:00:13 Hello and welcome to Week 2, Unit 4, Real-Time Replication with SAP LT Replication Server.

00:00:20 In this unit we are going to introduce the SAP LT Replication Server technology, show you the major use cases of it, also explain the architecture, how this fits into your landscape

00:00:30 or how you can integrate it into your landscape and we will give you a system demo to show you hands-on how this works in the system.

00:00:37 I think the SAP Landscape Transformation Replication Server is a well-known product

00:00:45 already in the world of SAP HANA native modeling. But since a few releases and especially with BW 7.4,

00:00:55 we are also able to use this technology to replicate in real time into the BW.

00:01:01 So in general SLT is an NetWeaver-based technology. It's working with database triggers,

00:01:10 means the tool, this technology adds a trigger on a source, on a table in a source,

00:01:18 and this table could be, it could be an ABAP-based system where a table could reside or it could be a non-ABAP means even non-SAP.

00:01:27 It could even be a stand-alone database but it has to be an SAP-supported database. And then SLT is able to create a database trigger and to fetch all the changes.

00:01:37 And that means as soon as there is new an insert, an update, an delete on a table happening,

00:01:44 this database trigger is writing, with a few other objects in background of course, this change into a target structure.

00:01:53 The target structure is, as we have seen earlier in the chapter about Operational Data Provisioning, the ODQ in this case.

00:02:00 In our case, the integration point between SLT and BW is definitely the Operational Data Provisioning framework.

00:02:07 There were also a few workarounds possible with earlier workarounds or technologies,

00:02:13 but this is really now a seamless integrated approach how SLT can write into the BW via this ODP technology,

00:02:22 and that means SLT is writing into the operational delta queue there, and from there we are fetching the delta and the data into BW.

00:02:32 So this enables basically a near real-time replication on database level from a database into BW.

00:02:39 It is the nearest real-time we ever had in BW. So before that we were...

00:02:48 our real-time definition was something around 3 to 4 to 5 minutes but with that we are actually really in real-time.

00:02:54 So, via real-time into the ODQ and from there it's the responsibility of SAP BW to fetch the data.

00:03:03 So you can actually choose when to fetch and how to fetch data out of ODQ to BW. You can do this in a scheduled manner using a scheduled DTP

00:03:11 or you can also use the RDA daemon or the RDA functionality of BW to continue with real-time processing beyond ODQ.

00:03:20 So this will be shown on the next slide but let's wait a second and really focus on the use case. So why should I do that?

00:03:28 First of all because we didn't have such a possibility before that for real-time replication into BW,

00:03:36 but this does not mean that we now replace all the existing BW extractors.

00:03:42 It rather means that we add a delta capability there where we do not have a delta provided.

00:03:49 I guess many of our customers know that there are a lot of, especially, master data or text data sources or extractors in ERP systems which do not have a delta.

00:03:59 Right... they cannot provide delta. But they reside there, may be generic extractors based on a single table.

00:04:04 So for these extractors it is actually possible to replace them by an SLT-based data source, which will then provide you with delta in a soon as...

00:04:13 And only the changed records will actually be transferred to BW. We do not have a delta criteria in a source like timestamp, like a date, like a whatever, a delta flag.

00:04:23 SLT adds this out of the box on top of the table. And this is a huge benefit especially when no extractor is offered.

00:04:32 And this is every time possible when we are facing database tables, then SLT can add a trigger or simple database views.

00:04:40 So it does not mean that we can replace complex business extractors like logistics, like financials,

00:04:48 because there is a huge logic happening in background or in a process. This is not possible.

00:04:53 But, we are really focusing on tables or views, and this is really the perfect use case then together with BW.

00:05:01 Okay, I guess we can now come to the technical or technical architecture.

00:05:06 When you look at how BW integrates this SLT sources, it's nothing else than, from BW perspective, as an ODP-based source system.

00:05:18 So far this is the new context. And in this context we will find the ODQ. This physically sits in the SLT server,

00:05:27 and the SLT server itself could be a standalone box or it could be integrated in the BW system like in our case here.

00:05:34 So everything which happens in the upper part of the picture here, everything which happens in the BW box is actually pretty standard, right?

00:05:41 You have a data source based on the new ODP technology but that is very similar to what we showed you in the ODP chapter for extractors. Looks very much the same. to use a data transfer process and a transformation on top of this, and you can do this either in a scheduled way or using real-time data replication.

00:05:59 We have the two aspects here: scheduled or real-time, it's depending all on the requirements.

00:06:05 But this is then the, what I mentioned as, it's in responsibility of BW how we then load the data out of ODP as a source. And this exactly what we are doing here.

00:06:16 The benefits are very clear. So first of all we are simplifying the data flow because, you see, with the ODP concept we can jump over the PSA table.

00:06:26 So, the PSA table is in this case optional because the data sits already in the ODQ and the ODQ is provided with real-time data.

00:06:35 We can have many subscribers, so it could also be that we share the delta with many BWs, so that they all receive the same delta information. It's totally handled by SLT.

00:06:45 And with that we are in less in a minute, so I think it's 30 seconds or something, the data will pop up in the BW system.

00:06:54 OK, so we are ready for the demo again. So let's jump into the system and have a look into the details.

00:07:03 So we start the demo with taking a look on the structure, on our target structure. You will find this here in an own SLT-related InfoArea.

00:07:15 There you find a DataStore object for sales orders. The sales order DataStore object is being built to store the sales orders out of the EPM model.

00:07:27 And then it's a standard data transfer, means we have a transformation, we have a data transfer process.

00:07:35 And the database, this is the special thing here, the database, no, the data source is then focusing on ODQ.

00:07:47 So, it's pointing to an ODQ, this is a transparent table in the source database.

00:07:53 SLT is fetching the data out of there, writing the data into the ODQ, and from there we take the data into the BW system.

00:08:01 So when you create such a data source from scratch, you would enter... Remember if you create an ODP data source, you have to provide the name of an operational data provider.

00:08:09 In the SLT context, this would just be the name of a table in the source system - table or a view.

00:08:15 In our case, we do it with a table. It could also be a database view. It doesn't really matter in that case.

00:08:23 There will be a trigger on the tables. If it's a view, we are triggering the source tables and then doing the logic on the fly within SLT.

00:08:32 That's basically what is happening. So this is our, so seen, interface to the ODQ SLT world.

00:08:39 You would see this is happening here an own context. So we have an own context which is SLT related for ODQ. We have an own source system.

00:08:49 And every source which is popping up from there is then based or supplied with SLT.

00:08:56 And the connection to the SLT world, it's an configuration on the SLT side.

00:09:03 This means a configuration opens up then a job in the source and constantly then fetching the data changes and write it to a target structure.

00:09:11 And since SLT is NetWeaver-based, we have an RSC connection. In a HANA native environment, it could also be a database connection to SAP HANA.

00:09:21 In our case we have two RSC connections for fetching the data out and writing it into the system.

00:09:26 And this is what we are having here in the demo, so we are using the same database in the same BW, but technically it doesn't matter.

00:09:34 And we are adding a trigger on the source and this is already there; we don't have to worry about this.

00:09:40 Okay. So let's see if we can see the effect of the data generation and really experience the real-time.

00:09:48 Right, for that we should start with having a look in the DataStore object first.

00:09:54 So for that we created a real-time query here, and if you execute this, you would see that the data out of the DataStore object is being fetched.

00:10:03 You see here a number of records which are stored at presence in the DataStore object.

00:10:09 And by the way, you saw the performance, which was for that amount of data not that bad. We are reporting directly in the DataStore object.

00:10:16 It's a standard DataStore object. The new advanced DataStore object will be covered in an own unit but from an idea it's basically the same in this case.

00:10:26 We are now focusing on this here. So today, we have about 327 data records being uploaded already. Now we create some delta records for that.

00:10:43 We are using the Data Generator by the EPM scenario. This is just a small shortcut for us.

00:10:52 You will find in the handout also the official, the more powerful generator. This is just a small one.

00:10:59 It's now generating sales order in background. Okay, there were now some sales orders... about hundred were created.

00:11:10 And if I now remember this one here, if I now refresh the query, it should, after some seconds, it should appear and you see now...

00:11:24 Just for explanation, the Data Generator it's generating sales orders but the amount of generated sales orders is different than the amount of delivered sales orders.

00:11:34 therefore the delivered... the items is higher. And what you saw now.

00:11:39 I think it were 15 to 20 seconds or something like that, it is definitely below one minute. It is definitely below one minute.

00:11:47 Technically we have in the DataStore object, we have an open request,

00:11:54 means there is always an open, so seen, area request where we can put the data into,

00:12:02 and then during run time, if you click here on Manage, you will see this one is open

00:12:09 this is our stream where we then can push data, and this is exactly the difference between the... if you know for example also real-time data acquisition technology before that.

00:12:18 This here is now pushing the data into BW. We had... We didn't created a process or something for that.

00:12:24 We just opened up the interface and then it's getting pushed in a few seconds into BW.

00:12:33 Okay, I guess that's it for the demo. Right. This is the demonstration, so let's just summarize again.

00:12:42 So, the SLT is really adding a benefit into SAP BW. This has also not much to do with SAP HANA.

00:12:51 It's related to NetWeaver. It adds real-time capabilities in data provisioning.

00:12:56 I mean so far, we load the data... or we didn't do much data loading so we were working in direct access mode with the data.

00:13:06 This was now a real-time data load into BW. In the next unit, you will also see some HANA-optimized ETL processes.

00:13:14 But this is really one benefit with the latest release of SAP BW, the delta capabilities with SLT.

00:13:22 The sources can be any table or views, it cannot be a standard extractor. If you would like to replicate the information of a standard extractor,

00:13:31 you have to focus on the sources of this extractor. Means you have to focus on the tables behind it or the view behind it.

00:13:39 And if this is a simple view like you said from master data, then SLT could rebuild this scenario.

00:13:46 If it's a very, very, complex scenario like a logistic cockpit, it will not easily possible. SLT can then be used in combination with SAP BW in two modes.

00:13:56 It could be scheduled via an DTP or it could be in real-time with the RDA technology with a constant push of the data into BW.

00:14:06 With the advanced DataStore object, we can also think about scheduling a data transfer process quite frequently, maybe even in a minute.

00:14:16 So this will be an advantage there. Due to the OTP framework, we do have all benefits available for this scenario, so we can have many subscribers and so on.

00:14:25 And we can of course jump over the PSA layer directly and write the data into the... in our case it was a DataStore object.

00:14:35 And by putting a query on top of this DataStore object, we were able to see new data popping up into BW in, let's say, 20 seconds or something.

00:14:45 So this was the real-time replication part, and in the next unit we will then speak about HANA-optimized ETL processes.

00:14:53 That's it for today. Have fun with your self-assessment.

WEEK 2, UNIT 5

00:00:12 Hello and welcome to week 2, unit 5: SAP HANA Optimized ETL Processes.

00:00:18 In week 1, we already touched a subject which is related to HANA optimized ETL processes, namely the HANA optimized DSO activation.

00:00:26 In this unit, we will mainly talk about optimizing BW transformations for SAP HANA.

00:00:32 and obviously, this will go along with a system demonstration.

00:00:36 So far in this week we saw already certain new functionalities and new innovations how we can bring data into BW.

00:00:45 And this is now more touching the traditional ETL processes and how we can optimize even those for SAP HANA usage.

00:00:54 Here's an overview of the optimization history in the ETL area.

00:00:59 We discussed in week 1, as we just said, the first step where we optimize the most expensive part in a classical BW ETL process, namely the

00:01:10 DSO activation, which we push down from the application server to HANA.

00:01:14 Now with BW 7.4 SP5 and SP8, we've made the next step of pushing down transformation logic.

00:01:22 So far, the transformation—you see that in the picture—was, until 7.3, still happening in the application server.

00:01:31 Only the activation of the DataStore object, of the whole data loading process, was pushed down to SAP HANA

00:01:38 and this was giving us a performance increase by a factor of 10. But now, also looking at the end-to-end data loading processes

00:01:48 allows us with BW 7.4—you see that—also to push down the transformation logic if certain criteria apply.

00:01:59 Obviously, if you use ABAP logic in a transformation routine, it's not possible to push this down to HANA.

00:02:05 Whereas, for example, all functions which you can use in a transformation,

00:02:10 with very view exceptions—I think we actually have a list which you can check on SCN—

00:02:17 all functions which are used in a transformation can be pushed down.

00:02:21 In general, or to keep it really simple, when you stay very close to the standard, meaning you are doing, for example, a master data lookup

00:02:31 in the standard rule, which is offered since 7.3, but nobody is aware of that, so everybody is writing the logic manually in ABAP.

00:02:42 And this caused that the transformation or the logic will still be processed in the application server.

00:02:47 But if you really use the standard transformation rule by the standard definition, we can push it down to SAP HANA and therefore we are able to optimized it.

00:02:57 Otherwise, the performance will be related in any exit—in transformations, maybe in planning exits, everywhere where you write ABAP code—

00:03:06 the performance is really related to how good or how bad your ABAP skills are.

00:03:13 And there's a final step of optimization, which is basically a very, very close integration with the SQL capabilities of HANA.

00:03:20 So it's actually possible—in a similar way to the ABAP expert routines which you are also able to use in BW transformations—

00:03:29 to now use HANA native SQL functionality in an expert-type mode of a transformation routine. We are going to show this.

00:03:37 We are using the so-called ABAP Managed Database Procedures for this,

00:03:42 which is basically a way to make use and leverage SQL capabilities managed by the ABAP framework.

00:03:51 So for example, things like transport, activation, will be handled by ABAP.

00:03:56 And inside an ABAP container, you are actually able to write native SQL HANA code.

00:04:02 From the BW perspective, this is the closest way how you can work, how you can write code with the database because it is obviously executed there.

00:04:11 It's written there in the SQL language, so this is really the closest integration.

00:04:16 This scenario was already shown in the openSAP ABAP on SAP HANA course where I will

00:04:26 shortly guest speak and show a system demonstration. And this is exactly the scenario. And one of the latest options we have with SAP BW 7.4

00:04:45 to integrate this logic via an ABAP Managed Database Procedure in BW.

00:04:40 So what are the advantages of this? Well, you have the full flexibility of SQL procedures, so nobody restricts you in anything here.

00:04:51 You also have much more responsibility because you're responsible for the correct type conversions. All this kind of stuff is now in your hands. But you have full flexibility.

00:04:58 And on the other hand, you also have full integration into the BW lifecycle.

00:05:03 So the SQL code which you write gets part of an ABAP Managed Database Procedure and thereby part of a BW transformation.

00:05:13 So the lifecycle management of the transformation basically encapsulates all the lifecycle management for the SQL code which you write.

00:05:21 This means this is another possibility. It does not mean you should write everything via an ABAP Managed Database Procedure.

00:05:28 Our guidance there is really stay as close as possible with the standard transformation rules because really actually they are optimized in background.

00:05:39 They are already optimized for SAP HANA usage—which I was not aware that this really happened until I was experiencing.

00:05:48 So this is really the way you should go and what we will demonstrate now in the system demonstration.

00:05:54 So just to prepare the demo which we are going to show in a second.

00:06:00 We are going to show you three different transformations. One will be a classical way including ABAP code.

00:06:09 Then we will show you what a similar transformation from a logical perspective equivalent transformation can look like if you use functions, purely functions.

00:06:20 And we will see that the system indicates that this transformation can then be pushed down and handled by HANA completely.

00:06:27 And the third step will be to show you the AMDP, ABAP Managed Database Procedure, functionality as well.

00:06:34 This is now really then also the first time we are now really physically also pushing data toward SAP BW and really leveraging traditional capabilities to store data in BW.

00:06:49 So maybe let's have a quick look at the example. We have here a specific InfoArea for this chapter.

00:06:57 So let's start with a transformation which is maybe the most classical one using a lot of ABAP code.

00:07:06 We are now focusing again on the sales orders. And taking the sales orders out of the EPM structure, pushing it toward BW.

00:07:15 And this is now the transformation happening in between.

00:07:19 The thing is what we demonstrated already in the first week, the physical way how the data is stored there is with different node keys,

00:07:28 which are there for an internal format. You see there are time stamps and especially these time stamps are very ugly to use for reporting.

00:07:36 So what we would like to do in this ETL process within SAP BW is really to take this time stamp and make it suitable for consumption by a date format.

00:07:49 Right. That's what this transformation rule here does. Maybe let's have a look at this in detail. Other than that, the transformation is quite trivial. Most of it is one to one.

00:07:57 But all of the time stamps in the source will be converted to a date format.

00:08:01 So let's see how this is handled in the most traditional way using ABAP code.

00:08:08 If you look at the ABAP code here in the transformation, you see the ABAP statement CONVERT TIME STAMP according to a certain time zone.

00:08:19 That's it, basically.

00:08:22 It's very simple, but it was not possible before 7.4 to do it in a different way because we didn't offer a standard transformation rule for that.

00:08:34 Therefore it was necessary to create this small ABAP piece just to transform the time stamp into a date format.

00:08:41 One thing you might notice in the GUI here. There's a new checkbox which says SAP HANA processing possible.

00:08:51 So once you activate a transformation, the system actually detects if it will be possible to push down this transformation for you.

00:08:58 So once you have created a transformation, you can see what the system indicates here.

00:09:04 In this case, we detect that there is ABAP code contained in the transformation. Therefore, the system will not be able to push this down.

00:09:12 There is also a Check button which allows you to find out more details about this.

00:09:16 If you are not clear about the reasons why this transformation cannot be pushed down, then you can use the Check button to get more information.

00:09:24 So let's do it HANA optimized.

00:09:26 Let's do it HANA optimized, right. So that would be here.

00:09:30 There are now coming two things together. First of all we see that we have HANA. We can push down the optimization.

00:09:39 And the other thing is that—as I mentioned with BW 7.4—

00:09:43 we have a new formula type available which offers us

00:09:50 to convert the time stamp into a date field by using standard transformation rules.

00:09:55 So here you see that the ABAP code is gone. There are no routings anymore, but there are formulas.

00:10:01 And these formulas basically have the same logic. If we look at the formula in detail, the name actually indicates what it does.

00:10:08 It converts a UTC time stamp into a local date according to time zone.

00:10:16 Now this is a logic which apparently we can rebuild in SQL style and therefore the system indicates that this transformation can be pushed down to HANA.

00:10:26 So what happens in detail is really you have to imagine that down in SAP HANA we offer now something, an internal model, an internal calculation scenario

00:10:37 which is really optimized to transform and to operate this transformation.

00:10:43 And this is exactly what we also will see in the runtime of the whole data loading process.

00:10:48 Maybe one thing. If you look at the corresponding DTP here, the DTP also offers you to actually decide if you want the system to execute the transformation in HANA or not.

00:10:59 So when you have an existing transformation which has been working in the application server

00:11:04 and you're not sure if you want to trust the automatic pushdown to HANA,

00:11:10 then by default the system will keep executing the transformation in the application server.

00:11:17 You can do tests by pushing down, compare the results, and once you're sure that the results are the same and that the transformation is handled correctly on HANA side,

00:11:26 you can actually switch the DTP to HANA Execution mode, and from then on things will be handled and pushed down to HANA.

00:11:33 Let's check the runtime. So what is the runtime of the non-HANA scenario?

00:11:43 Just go to the DataStore object and manage this one.

00:11:46 Okay, that's a good way.

00:11:47 Right. And then we can jump to the data transfer process. Just to the data transfer process. This is the easiest way, to the header.

00:11:55 There you see that it was taking us, for 13 million records, about 2 minutes and 3 seconds

00:12:03 to load data into the DataStore object.

00:12:07 Now let's compare this to the HANA optimized way based on functions.

00:12:13 We go to the Monitor. The result is significantly faster. So here it takes 51 seconds.

00:12:19 For the same amount of records. Exactly the same. The same logic, but just with the difference

00:12:26 that we pushed down the transformation, the main processing time, down to SAP HANA without having much logic happening in the application server

00:12:35 and it's usually...or not usually, but we saw quite often the factor too in optimizations like we see it here.

00:12:43 What basically happens is that we avoid a lot of data transfer between the database and the application server. And that gives you a significant performance benefit.

00:12:52 I think you remember the pushdown stuff, and this is really a great example of this pushdown logic.

00:13:01 Yeah, what I wanted to mention last was that the DTP logs obviously also indicate what execution mode was used for the corresponding request.

00:13:11 So you can even afterward check what happened here in the system.

00:13:14 And now let's go to the kind of expert mode where you work with basic SQL means to encode this transformation.

00:13:30 Maybe let's switch to the HANA native side for a second.

00:13:36 The DataSource on which all of this is built

00:13:41 is actually based on a database table...or maybe it's a view...called YSHEPM_V_SO.

00:13:51 That's the view which provides the data for this DataSource.

00:13:58 And if you do a select top 10, for example, of this view, you see that we have these kind of nasty time stamps from an analytic perspective in this structure here.

00:14:09 So what you want to do is basically write a certain SQL code which converts these time stamps into date formats.

00:14:18 And we've prepared something here. We've written a SQL statement which does this conversion for the three time stamp fields in the source

00:14:28 and keeps all the other stuff the same, except that it keeps track of the renaming between source and target.

00:14:36 What you have seen so far what we did in the first example in the ABAP routine,

00:14:42 then we did it via a standard transformation rule, and this is now we start to script it in SAP HANA. It's a very simple example, but it could be anything you script there.

00:14:52 It could even contain predictive logic, much more complex calculations. But this is another way to bring logic down to SAP HANA.

00:15:00 It's also nice to see how BW and the HANA native side are very close together because everything we did here was again in the Eclipse environment.

00:15:09 We used the embedded SAP GUI in the Eclipse environment to investigate the transformations.

00:15:14 Now we are using the SQL editor here to write and check a SQL routine.

00:15:22 And now we are basically...here you see the result, by the way. You see dates instead of time stamps.

00:15:28 And all we have to do now is basically cut and paste this SQL code here

00:15:35 into the transformation in BW. So we take this SQL code, snip it here.

00:15:42 We have to go to the Change mode of the transformation and we say we want to edit an SAP HANA Expert Script.

00:15:51 And now the ABAP Managed Database Procedure framework starts.

00:15:54 And we basically get a shell created by AMDP for us which contains some information about the source structure here and about the target structure down there.

00:16:04 And then there's a space where we can basically fill in our own logic.

00:16:08 So what we do here is just...okay.

00:16:11 That's because it's a transport, we're logging transports in the system.

00:16:18 And this is the same obviously when we work with HANA objects based in the BW, it's still the same logic.

00:16:24 So maybe select top 10 is not appropriate here. We want to select all data, right?

00:16:29 But all we have to do is we have to define the outTab, that's the output structure. We want to fill it as select this here from the inTab.

00:16:39 That's all we have to do here. Now we can activate this ABAP Managed Database Procedure.

00:16:44 And once we have activated this, we can go and activate the transformation.

00:16:50 And now, by the next execution for this transformation, we will execute the SAP HANA script in background which we've written on our own.

00:17:01 And you see also the check mark, SAP HANA processing is now enabled.

00:17:05 From a runtime perspective, we would still recommend to use the standard transformations as often as possible because there are also other optimizations possible.

00:17:15 But I think even this would be a better way to write logic than in the standard ABAP way.

00:17:23 Okay, so let's go back to the presentation here.

00:17:28 So what are the key takeaways of this unit?

00:17:31 First of all, SAP BW on SAP HANA significantly speeds up ETL processes. We've seen two cases,

00:17:37 basically the most expensive situations or most expensive process types in classical data flows: the DSO activation and the transformations.

00:17:45 Both of them can now be optimized and pushed down to HANA so that you gain a significant performance benefit.

00:17:52 We have, with the ABAP Managed Database Procedure, even a third way to write optimized code for SAP HANA via this script exit.

00:18:03 And I think therefore the end-to-end process has been optimized significantly.

00:18:09 I think now it's time for your self-test.

00:18:12 The next unit will show us more details about managing complex data warehousing processes where we will also focus about the new advanced DataStore object.

Week 2, Unit 6

00:00:14 Hello and welcome to Week 2, Unit 6, Managing Complex Data Warehouse Processes.

00:00:19 In this unit we are going to touch topics like data warehouse architecture, which is a very important topic in BW, in data warehouse implementation projects.

00:00:30 And we will show you two specific areas where BW provides services that make implementation of a data warehouse much, much simpler.

00:00:37 And that's on the one hand side what we call a smart data warehouse object.

00:00:42 It will be the advanced DataStore object, which really helps you with things like delta calculations and stuff like that but also query-specific things actually,

00:00:52 and the common scheduling and monitoring environment, which basically allows you to schedule all the processes and monitor all the processes

00:00:59 which you need to fill your data warehouse and work with the data which comes in and is distributed throughout the data warehouse.

00:01:08 Means so far, we simplified always the examples but actually data warehousing can be also complex.

00:01:16 And with SAP BW powered by SAP HANA we have already an application helping us to simplify certain techniques and approaches

00:01:26 but nevertheless, it's still necessary that, before the implementation start, that you actually have a plan in your mind and you know

00:01:33 how the final architecture or the structure of your data warehouse should look like.

00:01:39 And this is exactly what we would like to discuss: the data warehouse architecture.

00:01:42 Exactly. So, because data warehousing can be complex, it makes sense to think about architecture beforehand.

00:01:50 So, what we have there is basically... What you see on the right-hand side is a bit our reference architecture.

00:01:57 So in general, you should describe beforehand, and many people do this mistake

00:02:03 that they are not describing what they would like to do in the data warehouse or which areas they would like to have.

00:02:09 And this should be basically a document where you just provide some guidelines like,

00:02:17 hey, I would like to get my data as an inbound layer and then I do certain transformations.

00:02:23 And this is exactly what is reflected here in our reference architecture, which is called the layered scalable architecture

00:02:29 which also includes a lot of SAP HANA-specific things.

00:02:33 But to keep it for now simple, we should just focus on a few core areas.

00:02:38 So first of all, we have an area which is our enterprise data warehousing area where we do the core modelling,

00:02:45 means the harmonization, the consolidation of data.

00:02:49 Beside that, we usually would like to see in customer implementations that there is also a corporate memory area

00:02:56 and this is really helping you to keep all the historic values whatever you have loaded in your SAP BW system

00:03:03 because there will be requirements like, we would like to change a calculation but we don't have the data anymore in BW and staff like that.

00:03:11 So this is really your intelligent brain keeping the historic values like they were appearing in the data warehouse.

00:03:19 And the last area is the architected data mart area.

00:03:23 And this is the business-specific area where you could say okay, I have the financial guys in my company, I do have a controlling guy,

00:03:32 and both would like to have a different view on the data.

00:03:35 And therefore we start from a common set, from a single point of truth, like you learnt it already,

00:03:41 and then you do business-specific enhancement, and this is reflected in the architecture data model.

00:03:47 Right, so such an architecture basically provides a guideline when you implement a new scenario in your data warehouse,

00:03:52 what kind of layers you might want to use. Of course not always all layers are relevant but it basically specifies the relevant layers which are potentially useful,

00:04:02 and you can pick according to the needs of a given scenario.

00:04:06 What we also discussed so far is the bottom layer what we see here as an operational DataStore layer.

00:04:11 This is reflecting to our operational, to the Open ODS view where we are able to direct access external sources.

00:04:20 And beside that, we also have a agile area to model in a more flexible way.

00:04:25 But if you would like to have more information on that, we will also provide you, in the handout, a few links and information to see more about the layered scalable architecture.

00:04:35 One more thing about the open operational DataStore layers is, of course, also the ODP source system which we - right - discussed in detail.

00:04:42 The architecture is not everything because it will give you an idea how you should stack your data warehouse,

00:04:48 how you should layer the different objects you are using in the data warehouse.

00:04:53 But another important task is also to fulfil and to provide a common scheduled data processing across all the different objects.

00:05:03 And this is what we see here in the example, so the idea how you can load from InfoProvider to InfoProvider, and the framework managing this process.

00:05:13 I think in the picture on the right-hand side we already see two of the fundamental problems which we have in a data warehouse.

00:05:20 Namely, if you load between different containers, then obviously, if you have mass data, you cannot process the full data every time.

00:05:29 Data changes come in but you only want to process the changed data.

00:05:32 Therefore, it's very useful if these individual containers are able to computer calculate what data has been changed since the last run of a DTP or of a transformation.

00:05:42 So, that's one thing, one very common problem in data warehousing which can be solved in many ways.

00:05:47 BW basically provides a built-in way how to deal with this delta calculation.

00:05:52 And the other thing is that, that's something that we should see here, we have a look-up in the data flow,

00:05:58 which is here on the right-hand side, where we look up from the InfoProvider on the right-hand side.

00:06:02 So you need to make sure that the loads of the InfoProvider on the right-hand side happened before the load between the InfoProviders on the left side. Right?

00:06:14 And this kind of scheduling and synchronization of different loads, that is also something which is very important

00:06:19 because you want to automize this and not sit there manually and make sure that things run correctly.

00:06:25 You also have to make sure that when an error happens in one step, other steps are either not executed or behave accordingly in a meaningful manner.

00:06:34 You need basically two components for that: You need smart InfoProviders which we will cover later.

00:06:41 And you need a, so seen, framework for scheduling, and this is for us in BW, it's the process chain,

00:06:47 and on the next slide you see an example how such a process chain can look like.

00:06:54 It's kind of... It's very simplified, so I saw tremendous more complex things on customer side.

00:07:00 But usually it's the idea to have dependencies between data flows and to have one trigger.

00:07:06 And this trigger is really starting a whole data flow for example to do a daily load of all the finance data into the BW.

00:07:14 So here we have something where basically from the start, which is the trigger of the whole process, two different chains are, or two different load...

00:07:28 Two different data flows actually. Two different data flows, exactly, thanks for that. Two different data flows are just being loaded in parallel.

00:07:35 I mean this is very important also in combination with the architecture

00:07:41 because a good scheduling is of course helping the source system to define the load and the effort to process all the data in the BW system.

00:07:52 So typically, this goes along again with guidance. So you should avoid, of course, that you do reporting and data loading because it's influencing each other.

00:08:01 But again, with SAP HANA we have here a much more... a new situation because before that it was really critical to align all the different processes.

00:08:10 And this is now with all the things we've shown you in the data management area already and what we are going to cover soon,

00:08:17 it's much more uncritical than it was possible before because we do have also smart objects in BW.

00:08:25 And on top of the scheduling, of course, monitoring is always very important.

00:08:30 If such a process chain fails, basically one scenario for your businesses is maybe not available in the morning.

00:08:35 You want to be able to detect this, you want to be able to dive deep into the details where the error actually occurred.

00:08:41 And you want to have a tool which allows you to correct these errors and bring things together until business starts.

00:08:49 So now, let's come to the smart data warehouse objects and one of the most prominent features of BW 7.4 SP8, actually. That's the advanced DataStore object.

00:08:59 We announced it already in the early beginning, in week 1, that we have a new object for managing the data persistency in SAP BW.

00:09:08 And this reflected in this new object here.

00:09:12 And this goes along, of course, that we have to manage complex data warehousing processes with a high frequent number of loads every day, every hour, maybe even real time.

00:09:23 And therefore, we also re-innovated the traditional DataStore object which has been totally HANA-optimized.

00:09:30 So what we did in background again is really to implement everything completely new.

00:09:37 So we will see in this traditional DataStore object, for example, a new request handling, a new handling how we store the data.

00:09:44 But the services this object can fulfil are still the same. Actually, it offers more services because it consolidates existing InfoProviders.

00:09:53 Right. So the same for, of course, for DataStore objects.

00:09:55 So if you use a standard DataStore object for calculate a delta information or a write-optimized DataStore object for just a fast data loading,

00:10:04 this object is really able to fulfil all the services together in combination with an InfoCube.

00:10:12 Or in combination to take over even the InfoCube services. So this should really be and this will be our main object to store data in BW.

00:10:22 Right. So what's the, besides consolidation, what are the main advantages of the advanced DataStore object?

00:10:29 We already discussed in the chapter about open ODS views, that working with field data, working without InfoObjects is becoming more and more relevant.

00:10:39 And the advanced DataStore object basically can do both. So you can create an advanced DataStore object, 00:10:43 We will see an example in the system in a minute, which contains both, field information and InfoObjects.

00:10:59 You can really mix things and combine the best of two worlds.

00:10:53 We get rid of a lot of limitations of the past, for example the amount of key fields we could have in a DataStore object.

00:11:01 We enhanced this now to 120 possible key fields in this DataStore object.

00:11:07 And by that and by the ability to take over, for example, inventory handling and key figures, we are able,

00:11:13 and that goes along with the InfoCube consolidation, for instance,

00:11:17 so a whole new object which comes along with the possibility to switch between different behavior types.

00:11:24 We already mentioned the inventory handling which is going to be covered by this advanced DataStore object as well.

00:11:28 On top, planning will also be possible with the advanced DataStore object.

00:11:31 So itӳ really an object which combines the existing capabilities of InfoCube and DataStore object in one single object

00:11:39 and even goes beyond the existing capabilities of InfoCubes and DataStore objects.

00:11:43 And like you could imagine, the modelling experience. So what is the modelling experience? It's Eclipsed-based.

00:11:49 So, since this is a new object coming with SAP BW 7.4 SP8, we also introduce an Eclipse-based modelling environment for that.

00:12:00 But we also have to say what you're seeing here is now, at the time of recording, it's a development status

00:12:07 so it might be the case that this UI is looking like a bit different when you, for example, play around with that in your trial version.

00:12:14 But nevertheless, the idea and the concepts behind are definitely the same.

00:12:20 So what you see here, for example, in this Settings area, that's basically where you decide about the behavior of this object,

00:12:28 if it's more InfoCube-like or if it's more like a classical DataStore object.

00:12:33 Or even more generic: Do you need to calculate a delta information or not? Do you need to keep... Do you have outbound data from this. Exactly.

00:12:42 And by choosing the proper settings, you are defining in the end what type or what facet of this advanced DataStore object is been generated,

00:12:51 without that you have to know exactly why and what is the table behind that.

00:12:56 All the tables you saw also on the graphic, an inbound table, an outbound table, they will be generated in background by switching the services on and off.

00:13:05 Right. Actually, the nice thing is that even if you have an existing object with data loaded into it,

00:13:10 you can still modify these settings and change the behavior of the object. Whenever this makes sense, right.

00:13:17 So let's take a look on the details, on the field definitions. This is shown here in the Details where you see,

00:13:24 also based in our sales order scenario, that we have some field definitions in there but we also have some InfoObjects.

00:13:32 So this is again the purpose of this new object, to combine the field-based modelling world with the InfoObjects.

00:13:39 So you see some InfoObjects here, indicated by the InfoObject icons.

00:13:43 And you see some fields here which are basically just defined by data types as you know from the open ODS view.

00:13:51 Since this object is very, very new, again, and will be released with support package 8,

00:13:56 there will be also more and more features like the support of archiving, and we come to that in this course in the last week,

00:14:06 will be also be implemented in this object but this will be a first status.

00:14:10 And if you go to the Manage UI, I think this is also a big innovation, you will see that the old request handling has been changed dramatically.

00:14:22 What you see here is a new way how we display the different requests.

00:14:27 We have a different type of how we aggregate the daily administration.

00:14:32 You can see everything what was happening on this object, means it's keeping really a history of: the data has been activated, we deleted the data...

00:14:40 You will see every data movement. You see that the Managed UI is looking different, and I just can encourage you really to play around with it.

00:14:49 And again, keep in mind it will be looking a bit different but this will still be the area where you can activate the data,

00:14:55 where you can delete the data, work with the request.

00:14:59 And you do have much more information here in the monitor containing than with a standard DataStore object.

00:15:05 So let's summarize. This was the system demo. We did it already. So what is the key takeaway?

00:15:16 It's really important, like we said in the beginning, to have a plan, to have a reference architecture. And we are providing you with the layered scalable architecture.

00:15:26 It's also important to know that BW basically helps you to automize certain standard tasks like the delta calculation, or delta handling in general,

00:15:35 which is a common problem in every data warehouse project. BW has objects and mechanisms that do that out of the box and you don't have to care about it.

00:15:44 And it's very similar with the monitoring and scheduling capabilities.

00:15:49 All of this is also built-in into BW and everything fits together and works nicely together in order to give you one look and feel of the system.

00:15:59 And in the very end, of course, everything has been optimized for SAP HANA. We have new objects, we have aligned our reference architecture with that.

00:16:09 And now we wish you all the best for the weekly assignment. We see you next week.

www.sap.com