openSAP
SAP Business Warehouse powered by SAP HANA

WEEK 3, UNIT 1

00:00:13 Hello and welcome to week 3, unit 1 “Data Modeling Approaches with SAP BW powered by SAP HANA”.

00:00:21 In this week, and particular in this unit, we will now speak about new approaches in data modeling,

00:00:29 which are enabled with the latest release of BW and SAP HANA as a database.

00:00:34 We will talk about, basically – that are our two key concepts – the integration of SAP HANA models into BW, and the exposure of BW data into HANA.

00:00:45 And this comes since we see due to the combination of SAP BW and SAP HANA, we see a few trends also demanded by our customers.

00:00:56 One of the major trends is simplification. And I think we've seen certain instances of simplification already in the last week.

00:01:02 One of them was the consolidation of persistent InfoProviders where the advanced DataStore object consolidates things like the classic DataStore object and the InfoCube in one new object.

00:01:14 The same on the virtual types of InfoProviders where the CompositeProvider bascially is the successor of things like the InfoSet and the MultiProvider.

00:01:24 We've already, we've also seen topics like field-based modeling, which again is something where the advanced DataStore object plays a role,

00:01:33 but also obviously things like Open ODS fields.

00:01:37 We have talked about Big Data, federations, HANA smart access, and logical data warehousing in this context.

00:01:43 So this is again something which plays together with the Open ODS view, also the advanced DataStore object.

00:01:50 And now the trends which we are going to discuss in this week are the interoperability between the native HANA side and BW side.

00:01:59 And that's, as you said, in basically two directions: In the direction from HANA to BW, or from BW to HANA where we

00:02:08 are able to expose BW data models to the native HANA side. And in the other direction, consume native HANA models from BW.

00:02:18 And this is also perfectly reflected here in our reference architecture.

00:02:25 I mean, this has nothing to do now with the whole pushdown because the pushdown was happening in background with several objects.

00:02:33 Also in this week, we will see some pushdown functionalities, but this is really now about the modeling aspects of SAP HANA and SAP BW as data warehouse.

00:02:44 So what we would like to express here is really that if SAP HANA is the primary database for SAP BW,

00:02:52 we can, of course, also leverage the modeling capabilities within SAP HANA – means the SAP HANA analytical views,

00:03:01 the attribute views, the predictive libraries. And this is exactly how these operational data marts

00:03:08 because there is not that much of the lifecycle and these services around, get perfectly together with the data warehousing services.

00:03:17 With our architected data marts, with our entry layer in BW. And therefore, we are getting the best out of two worlds.

00:03:25 So that's really the message here. Don't consider these two modeling paradigms or these two different approaches, which are possible using SAP HANA as separate,

00:03:33 but really see them as a combination where both side benefit from each other.

00:03:38 Right. And that's the reason why we are considering BW powered by SAP HANA as, so seen, as a modeling platform.

00:03:47 And we have two strong use cases here driving us:

00:03:51 The first one is we have something in HANA. This could be an own, a native created model. It could even be an SAP HANA application – means an SAP HANA Live content, an accelerator.

00:04:03 It could be basically everything deployed on SAP HANA. And this can be combined with the BW world, and the integration object here is the CompositeProvider.

00:04:13 And as you see, that's something which we already described in, I think, week 1,

00:04:19 where we talked about the virtual types or the view types of InfoProviders.

00:04:23 And we already mentioned that HANA views and HANA tables basically behave a little bit like InfoProviders.

00:04:30 If you look at this picture here, in this example the CompositeProvider integrates BW models (depicted on the right-hand side), which look probably data objects here

00:04:40 as well as native HANA views. So it really doesn't make a difference when you're using a CompositeProvider whether the source or one of the sources of it is a native HANA model or a BW model.

00:04:50 And the big advantage maybe, it's been shown here in the short example a bit better,

00:04:56 is really that with this approach, we can add, on top of an SAP HANA native environment, the BW Analytic Manager – the OLAP Manager functionalities.

00:05:06 This will be shown in various examples during this week, but there's one important thing also to mention:

00:05:15 You cannot combine everything with BW, so the one or the other now asks us: “Hey, can I deploy then my ERP, my ECC instance next to my BW?”.

00:05:24 Now this is not possible. So we are speaking here about native modeling, native content on HANA views, HANA tables, but not an ABAP...big application.

00:05:37 And this is one of the use cases we will focus during the week.

00:05:41 The other one is the other way around. And this is, we have something in BW. We have our data there, maybe stored an InfoProvider, maybe stored in

00:05:50 master data, in InfoObjects. And this master data, for instance, should now be used in SAP HANA native environment,

00:05:58 because I have my master data already there. Now I'm opening up the BW world and add the SQL flexibility of the SAP HANA environment on top.

00:06:08 Now, the important point is here that we do not just open up the BW tables to native HANA consumption,

00:06:13 but we generate HANA models which encapsulate all the dependencies between, for example, the master data attribute and text tables,

00:06:22 or which also include information about hierarchies. So all of this, all this information which is contained in a BW model

00:06:30 is basically compiled into a HANA native view which is exposed for native usage.

00:06:36 So what you'll get on HANA side is more than access to the flat or plain database tables,

00:06:42 but actually something which already contains all the information which you have modeled actively on BW side.

00:06:49 And why are we doing this, why this approach? And again, this will be one unit, because it would not make sense to create a HANA view directly on the BW source table,

00:06:56 because you should leverage the official interface, we have all the authorization and stuff like that.

00:07:01 And if you go to the example, you will see the interesting part here is that via this approach we are getting open for a SQL-based consumption.

00:07:11 And this is really new also for us in SAP BW because via this approach we can reuse, or we can use, third party tools,

00:07:20 any applications, any SQL-based tools also from SAP. And this is what we are going to demonstrate based on SAP BW data in the HANA environment.

00:07:31 So SQL-based Business Intelligence tools are one example. Of course, all the other HANA engines, which you have in your HANA database,

00:07:40 are also thereby opened up for working with BW master data, for example, or BW transaction data.

00:07:48 That's a very important aspect as well. So it's really opening up the value of your BW data to many new use cases.

00:07:55 And if you now ask yourself, I'm confused, when should I do what? This is kind of a collection of aspects. So we will not cover all of them, but you can do this in the handout.

00:08:08 It's also just, you know, an aspect list. It's not a “right” or “wrong”. It's just expressing: “Hey, in the SAP HANA world, I'm really used to use my SQL knowledge.”, for instance.

00:08:20 In BW, this is an ABAP application, so I have a strong application server to write, for instance, my own ABAP logic.

00:08:28 I have a whole lifecycle environment with our Transport Management and stuff like that.

00:08:33 And this comes on top and in combination with my flexible HANA techniques.

00:08:39 Think of all the things in BW like data lifecycle management, near-line storage. All of this is a strength on BW side.

00:08:47 On HANA side, things are more lightweight, more flexible, more agile maybe. And really the combination of both brings the most value.

00:08:55 This is exactly how the whole week here is tailored. So we will show you, and this is what we expressed here already, the major trends in the data modeling,

00:09:04 what is really now new with SAP BW 7.4 with SAP HANA as a database.

00:09:09 We will talk about the so-called mixed scenarios by leveraging the HANA world with BW.

00:09:15 We will demonstrate how you can very easily consume SAP BW data with SQL due to this HANA view generation, and we do the other way around.

00:09:27 So you will see how you can leverage, for instance, HANA predictive libraries in an analysis within SAP BW.

00:09:35 You will see how can even use a planning process which has input data from the HANA native world.

00:09:42 This is really what we would like to show here and really what the strategy is. So the combination out of SAP HANA with SAP BW as application.

00:09:52 is here the key aspect. And we will start in the next unit with the new CompositeProvider, which has nothing to do with the old CompositeProvider.

00:10:01 And now you can enjoy your self-test.

WEEK 3, UNIT 2

00:00:11 Hello and welcome back to week 3, unit 2 “Simplified Data Modeling with the New CompositeProvider”.

00:00:20 In this unit, we will talk about the new CompositeProvider with SAP BW 7.4.

00:00:26 We will talk, we will see the modeling experience in the Eclipse-based environment for the CompositeProvider.

00:00:32 And we will touch all the different use cases enabled with the CompositeProvider.

00:00:39 So, what's really new about the CompositeProvider?

00:00:42 From my perspective, the most important thing which you should take with you is that we now have one single object for the whole virtual Data Mart layer.

00:00:49 So all kinds of composition, either union or join, whatever. What you used to do with MultiProviders or InfoSets is now combined in one single object. Right.

00:00:58 I think that's a huge advantage since BW 7.4. It's a simplification what we talked already about.

00:01:05 And on the other hand, the CompositeProvider was already there with BW 7.3, but this is really now a renewed object.

00:00:13 It's optimized for SAP HANA, so in background in the database we have an optimization model

00:01:20 for each CompositeProvider, so this is really the object in the area of data composition.

00:01:26 Every time it's about to join data, the CompositeProvider will be in the future the object and the architecture.

00:01:34 Before we come to the details, one aspect which we should also mention is, I already said that,

00:01:39 when it comes to composition of BW objects, the CompositeProvider is the tool.

00:01:43 It's not only BW objects but it's also native HANA objects. So all kinds of HANA views we will see will also be supported.

00:01:49 When we talk about the data modeling approaches in the previous unit, the CompositeProvider is exactly the object to integrate SAP HANA views in BW,

00:01:59 and offering different supported join types, which was not possible before that in one single object in BW.

00:02:08 We have a long history with of this kind of composition operations, of course. If you are probably familiar with the MultiProvider for unions and the InfoSets for joins.

00:02:17 Now both of these options are basically merged in one new object: the CompositeProvider.

00:02:23 It supports, as we said, unions. It supports inner joins and left outer joins. And it supports all these operations on a whole variety of source objects.

00:02:32 So the source objects can be basically everything you know from BW, as native BW objects.

00:02:38 It could be every InfoProvider, it could be SAP HANA models, related master data objects.

00:02:46 So this is what you really can do with the new CompositeProvider since BW 7.4.

00:02:52 Last not least, the advanced DataStore object. So also the very new objects are already supported by the CompositeProvider.

00:03:00 The strategy is really clear on that, on that situation, so the CompositeProvider is really the main object for this kind of operation.

00:03:08 And supporting the different use cases for integration of – and that's use case one – of BW objects,

00:03:17 so pure BW use case replacing virtual MultiProviders and InfoSets.

00:03:24 So it can be any InfoCube together with a DataStore, with an advanced DataStore object, with a semantic partitioned object. So this is our use case one in a BW.

00:03:34 Yes, use case two is combining BW data with native HANA data, with SAP HANA views.

00:03:40 We've splitted it here into use cases, but you will see in the system demonstration later that it actually feels very much the same.

00:03:46 And that again is something which we already mentioned in one of the earlier units

00:03:51 that SAP HANA views and SAP HANA native objects actually in many ways now very much feel like BW native objects, BW objects in the system.

00:04:04 We're replacing with use case one objects which have been there since BW 7.3 in combination also with the Business Warehouse Accelerator,

00:04:13 which is the TransientProvider, the VirtualProvider based on SAP HANA models. So this is use case number two

00:04:19 where you can directly in a search popup search for SAP HANA objects and integrate it in the CompositeProvider.

00:04:26 Use case three is basically the other extreme: You only work with native HANA objects, but you want to leverage BW OLAP functionality on top of that.

00:04:35 And the CompositeProvider also supports this. So all the operations which you can do with BW objects can also be done with native HANA objects.

00:04:44 Right. And this is helping you to have, so what we described here as a single point of truth for one Access layer for all the data.

00:04:53 So many customers demand really to have one defined Access layer for corporate reporting, no matter where the data sits.

00:05:01 And the CompositeProvider could be very much this layer, or it's really our object to add BW authorizations, BW services on top of SAP HANA views.

00:05:13 So I guess it's time for the demo already.

00:05:17 We want to jump right into the system? Right. We will show the different use cases we have with the CompositeProvider.

00:05:22 We will talk about the optimized Eclipse-based modeling experience. And this is exactly what you see in the BW modeling environment.

00:05:32 So here you go. What you see here is our info area for the CompositeProviders. It's also there in the HANA, in the cloud version.

00:05:43 And this is the place to create a new CompositeProvider.

00:05:47 With BW 7.4 we are connected to our back end here, and just create a new one.

00:05:55 We call it “YEPM_DEMO”...demonstration, so.

00:06:04 And now you see here already the different options for join types. You see that we can now select between a union or a join.

00:06:13 And if we select “join”, we have the options to do an inner join or to do a left outer join.

00:06:19 And again, this combined in one object was not possible in BW before, and therefore it's a great benefit here.

00:06:26 What we will do now is we will model our – if you remember in the first week – our simplification case

00:06:33 where we spoke about how the architecture is really changing with BW 7.4 powered by SAP HANA.

00:06:41 And this example – means a union between two DataStore objects – is exactly the use case we would like to model now.

00:06:49 So what you see here in the first scenario on the first tab here is, are a few properties, but they should not bother us that much.

00:06:58 We jump right into the Scenario tab. And this is the area where you can define the join.

00:07:04 Here on the left-hand side, you see the join note. And this is the place where you can add different part provider.

00:07:13 We take now a classic DataStore object. It could also be an advanced. This is our DataStore object for the open sales orders.

00:07:20 We take this into account. You see now we have the source structure enhanced with the fields by the first part provider.

00:07:29 Let's add another one...And this is the DataStore object for the closed sales orders, means our already delivered ones.

00:07:40 And if you click on the union note here, you see now all objects available to define this union.

00:07:49 And this is simple, a very simple drag and drop experience, so just take the source objects and assign them to the target structure.

00:07:59 And we see – and it's really also for us in BW very nice – you see in a graphical way that we assign now the source objects here to the target of the CompositeProvider.

00:08:12 And this is how the join is defined because, or the union is defined, because we are using a union for two identical structures to each other.

00:08:19 And you directly see the assignment of source InfoObjects to target InfoObjects here.

00:08:25 Obviously, these lines here are suggestions by the system. In this case, it's trivial because the structure of both source data objects are identical.

00:08:35 And therefore, the target structure is clearly the same, and the mapping is also trivial for the system. Right.

00:08:40 If you do not like the graphical model experience – what I Do – you can also switch to the form-based, so you see the table and with

00:08:48 the different assignments, but I think the graphical one is, for me, much, much nicer.

00:08:54 The last tab is here the Output tab. And there you see the output structure of the CompositeProvider, and you can do there different kinds of assignments

00:09:04 if the system was not able to assign already an InfoObject to it.

00:09:09 It also can work, of course, with fields. So an InfoObject is not mandatory here, it could also be a join of two

00:09:18 field-based objects, and then it's working like exactly the same.

00:09:24 And the last thing here is that you can then assign certain properties for the query runtime like

00:09:31 how should the InfoObjects be displayed – should it be text or key?

00:09:36 So if you look at the right-hand side here, it looks very similar to what we've seen in the Open ODS view.

00:09:41 That's very much aligned because it's dealing with a similar problem. Therefore the solution should look very similar and both cases are very much, very much the same.

00:09:50 Right. And let's activate it. So now the CompositeProvider is active.

00:09:58 You would now see this CompositeProvider also in the ABAP back end, an RSA–1.

00:10:03 It would be also displayed there. You could also expand it and would see the part providers.

00:10:09 But the change is exclusively possible here in the Eclipse world. You could not change it in the back end.

00:10:16 OK, but it's, I guess, everything is ready to run a query on this object now. So let's jump into Analysis from Microsoft Excel,

00:10:26 and open the default query on this CompositeProvider.

00:10:31 And you see, well, with a maybe moderate data volume of roughly 7 million records, the result was there instantly.

00:10:40 Now what happens, for example, if we drill by InfoProvider, then we expect to see basically the contribution of each of the source DataStore objects to the total result.

00:10:51 That's what you can see here. Another thing which we could do is maybe drill down by order currency.

00:10:58 You see now already the stars are indicating a sort of OLAP intelligence, so it was not possible to sum up all the currencies without a currency key.

00:11:09 And this is something we will also talk about in the, we have an own chapter for the OLAP functionalities where talk about especially these scenarios.

00:11:19 This functionality which was now possible between two InfoProviders would also be possible if you take an InfoProvider, and, for example, combine it with an SAP HANA model.

00:11:30 Yes. So let's have a look at such an example. We've prepared something in the system for you here.

00:11:37 Let me open the info area. Here for example, we see a CompositeProvider which does a join.

00:11:46 So here we have an inner join of a DataStore object with an InfoCube.

00:11:50 And, well, on the left-hand side you clearly see the logic how the individual part providers are going to be combined – it says “inner join”.

00:11:59 And the details can be seen on the right-hand side. So here you again see the, or you could expand this and see the explicit structure of the source objects.

00:12:07 On the right-hand side, one of the most important information in this case is probably the join condition. You want to see how do I,.

00:12:14 you have to specify, and afterwards you want to see and maybe modify how the join conditions actually are maintained.

00:12:19 So that's what you see here on the righ-hand side, but the rest is very much the same.

00:12:24 I mean we have mappings, we have a target structure, we see the mapping from the source objects to the target.

00:12:29 And of course, on the Output tab, we would again be able to specify details for each of the objects in the target structure.

00:12:37 And in this case, it was again the two BW objects, so DataStore object and InfoCube.

00:12:42 And if you go to the, I think we have also another example here, where we do basically the same.

00:12:49 But now we combine...one HANA model – means that's a calculation view modeled in the HANA native environment.

00:13:00 And we join it here together with the BW DataStore object again for sales orders.

00:13:08 You see the modeling experience is pretty much the same. And so also the reporting experience. It's from BW perspective, from the consumption perspective,

00:13:18 the CompositeProviders is really then the entry layer, the integration layer, and it adds up all the functionalities you would expect from a BW,

00:13:26 and therefore can combine the HANA native models here.

00:13:30 Yes, just to stress what you just said. If you don't look explicitly or concretely at this icon here,

00:13:37 you would probably not even be aware that this is not a BW object but a native HANA view, a calculation view in this case.

00:13:44 So the whole modeling experience is really the same, whether things come from BW side or from native HANA side.

00:13:50 If you ask now yourself how is it possible – native HANA and BW – this has been explained in the data modeling approaches,

00:13:59 where you in the handout also find the different recommendations and the so seen also restrictions about what can I combine together in one harmonized database, and what's not possible.

00:14:11 Yes, this was the new CompositeProvider. It's our main object for all the join and union operations in BW.

00:14:20 It provides that in a very, very flexible way. It's a graphical Eclipse-based modeling experience, I think that's

00:14:28 a very new experience for us in BW. It's a really nice modeling approach.

00:14:34 The CompositeProvider consolidates existing InfoProviders like we talked and we showed also in the Simplification unit

00:14:41 with VirtualProviders, InfoSets, MultiProviders.

00:14:45 And if you ask us now: “What should I use? Should I use a MultiProvider, or a CompositeProvider in future?”,

00:14:50 the answer should now be very clear that the new CompositeProvider – since BW 7.4 – is offering the functionality.

00:14:57 It's optimized from a performance perspective with SAP HANA. It's an exclusive feature there, and therefore this is the feature to go.

00:15:07 So this was the first use case of the data modeling approaches to integrate SAP HANA models into SAP BW.

00:15:16 And in the next unit, we will show the way around, means how can you expose BW models into the SAP HANA native environment.

00:15:26 But before that, I guess it's time for the self-test again.

WEEK 3, UNIT 3

00:00:13 Hello and welcome back to week 3, unit 3 “SAP BW and SAP HANA Interoperability”.

00:00:20 In this unit, we would like to focus on how SAP BW models can be exposed to SAP HANA.

00:00:27 And what we will show there is how you then can leverage the SAP BW data in an SAP HANA environment.

00:00:34 And we will then show a demonstration and consume the data with SAP Lumira.

00:00:39 So let's go over and discuss the technical details about this a little bit.

00:00:43 So what you can do with this functionality is you can leverage your data in BW, which could be

00:00:50 master data or transaction data. So you see a DataStore object here as an example. And you also see master data InfoObjects on the right-hand side.

00:00:57 And what BW does for you as a service is that it generates native HANA views based on the metadata information it has,

00:01:04 which basically means that we have a way to consume BW data, for example, via SQL for front-end consumption.

00:01:16 Any other SQL engine which runs on your HANA installation can now leverage BW data. So we really open up BW data to a whole new type of consumption.

00:01:28 It's not only SQL consumption by the way; all other HANA engines are also able to operate on BW data in this way.

00:01:35 Like predictive, like a text engine. Everything which is known by SAP HANA can reuse these structures, of course.

00:01:43 And the key aspect is here that we are really taking everything important from the SAP BW side into account.

00:01:51 That means we are taking the authorizations, everything which is necessary to rebuild a DataStore object, an InfoObject in a HANA native world.

00:02:00 And we are generating this into the SAP HANA schema. And we'll see in a second in the system demo that this is more complex than just using the plain tables of these objects.

00:02:10 You'll see that, of course, you need to understand the relation between different tables.

00:02:15 And all this knowledge is, of course, contained in the BW model. And BW compiles this into

00:02:22 the native HANA views, and therefore you get a very rich view on HANA side, which gives you all the information which you have on BW side.

00:02:29 And this is really the way how you should consume BW data in SAP HANA.

00:02:34 Because to consume the data directly on table, this is not supported and totally, absolutely not recommended.

00:02:43 Because then you are not leveraging all the authorizations, all the conversions. So this is really the interface to write a SQL statement towards SAP BW data.

00:02:54 So let's come to the details about the objects which support this kind of feature.

00:02:59 Of course, all the transaction data containers which we know – the InfoProviders like DataStore object, also the advanced DataStore object – supports this feature.

00:03:09 InfoCubes. The CompositeProvider does it as well. So even more complex things than individual InfoProviders. Also the joins, which you can model in a CompositeProvider.

00:03:19 Also these can be exposed to HANA native usage afterwards.

00:03:23 And I think also our master data object, also the InfoObject can be generated as an SAP HANA view.

00:03:30 And this will contain, of course, all the tags, all the attributes, everything which is important. Beside in hierarchy this would not be possible to physically expose it towards an SAP HANA model.

00:03:43 Right. Maybe that's was mentioning because I made this, and I apologize for that, I made an error I think in unit 1 this week, where I mentioned that hierarchies are also supported. Hierarchies are out of scope in this feature.

00:03:54 Right. So this is really our way how we can extend the SAP BW model with SAP HANA native models.

00:04:02 You can reuse the harmonized master data, and this is exactly what we would like to show here in a short demonstration.

00:04:08 So we will generate an SAP HANA model one time for transactional data, and on the other side we will take an InfoObject like in our first example here.

00:04:18 So we have the EPM DEMO InfoObject and – for the Business Partner. And there you can see the property of SAP HANA model generation.

00:04:29 Yes. So we're here in the SAP GUI looking at the InfoObject 0D_NW_BP.

00:04:37 If you go to the Master Data and Texts tab, you will see that, if you scroll down, there's a new check box which says External SAP HANA View.

00:04:47 And if you check this box here, you will see that this has an influence on the native HANA side.

00:04:57 So if we go to the HANA Adminstration Console here, and open the Content folder, which contains all the native HANA models –

00:05:01 the analytic views, attribute views, and calculation views in particular.

00:05:10 There's a specific package, which is called “system_local”. And under this package, you will find a package called “bw”.

00:05:18 And in another subpackage, which is called “bw2hana”, we'll find attribute views which are generated...

00:05:26 Or in this case, the attribute view which is generated from this InfoObject.

00:05:31 Actually, it's more than just an attribute view. We also generate an analytic view for an InfoObject because InfoObjects also in BW play a dual role.

00:05:39 You can use them as master data objects, but you can also do master data reporting. So they can also act as InfoProviders.

00:05:44 And once an InfoObject has this check box that it's also – we actually see it here that the characteristic is also an InfoProvider,

00:05:52 then such an analytic view is also generated, so that you can do reporting, leveraging key figures contained in the master data.

00:06:01 And this path where we stored the data here in the content area, this is also you can maintain by yourself. So far, we took the default proposal, which is “bw2hana”.

00:06:11 And what you see when we open the attribute view, for example, is really that this is now a pure SAP HANA model, which is nothing else than an attribute view.

00:06:21 And if you go into the data foundation, and then we make the tables a bit nicer, you will see that

00:06:26 all the tables which are necessary, all the text information, all the language-dependent things are really contained and generated automatically from BW

00:06:39 during the InfoObject activation automatically into this object.

00:06:43 So you see that this model here not only contains – here it's actually the P tables, so the time-independent attributes of this InfoObject –

00:06:52 but also the master data, or in this case the text, of attributes of this InfoObject, right?

00:06:59 So here we have the text table for the Business Partner, for example, which is also compiled into this model.

00:07:05 We have the text table for the country. All of these, so basically all the additional associations which come from attributes are also compiled into this model.

00:07:15 And you see that it would be very tedious and very difficult to build such a model on your own.

00:07:19 You would really have to understand all the dependencies between all the different tables of different InfoObjects if you wanted to build this on your own.

00:07:27 And like we said, we take here every authorization, everything in background. We generate HANA privileges

00:07:35 for the BW analysis authorizations for specifically this model.

00:07:40 And if there is a change happening with the BW object, and we activate it, which will basically happen every time after a transport, or a manual change, or whatever,

00:07:49 we regenerate this, and we regenerate also the user privileges and stuff like that.

00:07:55 You also see that we take care of the technical fields, like here is the object version.

00:07:59 You might have active or changed master data. Of course, this model contains a filter which only shows the active master data.

00:08:08 So all this is basically handled for you and you don't have to care about it. All you need to do, or what you can do, is just leverage this generated model.

00:08:14 And therefore it's so important that consume this directly. If you look now to the DataStore object,

00:08:20 you will see that this is even much, much more complex because a DataStore object or maybe an InfoCube really has a lot of tables in background on database level.

00:08:31 And if you just consume one table, like the active table on the DataStore object, you will not get the whole data view.

00:08:38 There will be information missing. And here we see again – it's a bit similar to the example before –

00:08:45 all the information which comes through the attributes here and the other characteristics in this DataStore object,

00:08:51 all the texts, in principle also master data, which is joins or navigation attributes, would be compiled into this analytic view.

00:09:01 And you would basically get the whole star schema as BW leverages it by itself.

00:09:06 Right. And this is then the starting point to do reporting. Like we said, two things you can think of:

00:09:13 You could reuse this InfoObject, this DataStore object, whatever. You generate it into the SAP HANA world for modeling there.

00:09:21 So you could even model now an own calculation view, which is based on your attribute view.

00:09:27 You could do predictive analysis, whatever you would like to do with this generated model.

00:09:33 Or you consume it via a SQL-based tool because this is now an interface coming due to SAP HANA.

00:09:42 We will take now our example SAP Lumira as SQL-based tool, and connect directly to the SAP HANA system.

00:09:52 OK, so let's see how you do this. Well if you're not familiar with Lumira,

00:10:00 what you have to give here is, of course, the connectivity to the system. We've prepared this already.

00:10:05 And this is the, to clarify, it's really the HANA side now. So you need, of course, then also a user in the HANA side that you can consume the data directly there

00:10:15 because what's happening now is that SAP Lumira is sending a SQL statement to the HANA database directly, so BW is not participating at all.

00:10:23 Beside the fact that in the end the data is coming from the BW schema within the SAP HANA database.

00:10:30 So now we connect with a user. We will then...

00:10:37 SAP Lumira is then fetching all the different views which are available in the system.

00:10:42 We take now one of our sales order DataStore objects we used already in previous exercises. That's actually the one we just saw in the background.

00:10:53 Take this one. We also have here a key date for time dependencies available. We are not using this, so let's

00:11:01 keep it empty. And then what Lumira is automatically doing is preparing here the output structure.

00:11:09 And we can now work and...It's a tool for ad-hoc analysis. So we can take now, for example,

00:11:17 we can take a key figure, put it on the measure side here. Now you see the overall amount, which is dramatically high because it's sample data from us.

00:11:25 Then we take, for instance, the country. And see it's a really ad-hoc analysis what we can do here. We can change even the chart type.

00:11:35 We can add maybe a pie chart. So we can do several things, which is really a nice tool for

exploration of the data with ad-hoc analysis capabilities.

00:11:49 And one thing which is also very nice is we have here a country information. And you see that we could also add the

00:11:57 geographic information here. And this is exactly what we also could do. In our country information, we have an abbreviation for each country.

00:12:05 And SAP Lumira is able, by adding now geographical data detail, to assign this country information to

00:12:16 a country information which is coming from SAP Lumira. OK, we say 19 times found now an assignment between the data which is coming from us and the data which is there.

00:12:28 And if we create this, we get a new geographical detail here. And now we can even out of the box, without that we did now much preparation. And beside we are not front end guys.

00:12:41 So it's very easy for us even to now take even the geographic information in here, and you see

00:12:48 we're getting now a map here with all the different details and can see, per country, how much the value of our sales orders was there.

00:13:00 I guess that's it for the demo today. So let's wrap up what we saw.

00:13:07 Well you've seen that there's a big value in using these generated views from BW side.

00:13:14 It's much, much easier than even thinking about creating such views on your own.

00:13:18 Besides, the system takes care of all things like transport, lifecycle management, all of this. And it knows the dependencies between all the objects involved.

00:13:26 So that's a very, very powerful service. And it opens up BW data to basically any type of consumption

00:13:32 because SQL is basically the most common interface for any kind of tool, front-end tools, whatever.

00:13:39 And thereby BW data is basically usable with any engine or tool you can imagine.

00:13:45 Mean it would also be possible to write a manual SQL, so you could query manually on the Console level your InfoCube data.

00:13:53 But what you just saw is really how easy it is if you have a very nice tool like SAP Lumira to add a reporting on SAP BW data,

00:14:04 which is, in the BW world, also quite new because these are really possibilities also in a speed which is, which was not possible before.

00:14:15 OK, that's it for today. Right. The next unit will cover the capabilities of the Analytic Manager in SAP BW,

00:14:22 which is a very, very important part also in the combination between SAP HANA and SAP BW.

00:14:30 And I guess it's my point to remind you of the self-test again.

WEEK 3, UNIT 4

00:00:13 Hello and welcome back to week 3, unit 4: Leveraging the BW Analytic Manager.

00:00:20 In this unit, we would really like to clarify what exactly does the Analytic Manager, also called the OLAP engine of BW really do.

00:00:29 What are the functions offered there? How is the collaboration with SAP HANA? And really how is the processing looking like?

00:00:36 In general, the BW Analytic Manager is really kind of a bit the brain of BW.

00:00:42 It's really where the complex calculations for reporting are happening, and there are various components playing together to fulfill this service.

00:00:52 So if you look at the picture on the right-hand side here, the stack on the right-hand side, on top we see the BW or BEx Query Designer, which is basically the design tool for analytic reports in BW

00:01:03 and thereby also the design tool for the functions offered by the Analytic Manager.

00:01:08 So what you do in the Query Designer is basically you design the layout of a report. So you decide which of the characteristics to use for drilldown,

00:01:16 which of the key figures to use, typically in the columns.

00:01:20 Maybe you have something like exception aggregation, certain filters associated to key figures as well.

00:01:27 All this definition is described and modeled in a model-driven way in the Query Designer.

00:01:33 There's no SQL involved, or no programming language. It's really a graphical way of modeling a report.

00:01:39 And this report definition is then interpreted by the analytic engine.

00:01:44 You have to imagine that the analytic engine is really translating this complex product of definitions, of calculations, toward a database, in our case toward SAP HANA.

00:01:56 And, of course, we optimize this translation in this case. So we are natively pushing down this complex logic, these definitions, down toward SAP HANA.

00:02:07 We are leveraging the various engines we have available in SAP HANA. And therefore, we have this excellent query performance with SAP HANA.

00:02:15 So what we did so far was really we were consuming, in the previous unit, directly data from BW via SQL.

00:02:23 But if it's getting more complex, if you really would like to define queries for a specific business purpose with complex calculations with definitions,

00:02:33 then we recommend to build this query via the Query Designer. And I think this is now what we should also shortly take a look.

00:02:43 We are pretty sure that the master view where how the Query Designer looks because there was not much change happening the last years.

00:02:52 So here's a quick look at the Query Designer.

00:02:56 Down here you see a preview of the result will look like. So you always can check what you will have to expect from the current report.

00:03:06 In this example, we have assigned the product ID and the order currency to the rows.

00:03:12 And you can actually see this here, that we will see groups for individual products and subgroups for certain order currencies.

00:03:23 And we also see the key figures which we have assigned to the columns here.

00:03:27 The key figures are actually quite tricky in this example because they are not described by something which is directly contained in the underlying InfoProvider,

00:03:38 but they are actually calculated as formulas on top of that. So it's not just aggregating numbers which are contained somewhere in the database,

00:03:45 but there might be some pretty complex calculations involved.

00:03:50 There could also be additional filters like you might sum up the sales order volumes only for the current year or for the last year,

00:03:58 or you might have two key figures in two different rows—one for the sales order volume of last year compared to the current year. All this kind of stuff could be done here.

00:04:06 So it's really the definition of a key figure can be pretty complex.

00:04:09 Right. And this is all, of course, happening on the BW data, so this based on an InfoProvider and then can be consumed by reporting tools directly

00:04:18 but they query really encapsulates all this complex logic, all these definitions for consumption by the front-end tools.

00:04:25 Maybe one more thing about the BEx Query Designer for those of you who know the tool and know that it has been around for quite a while.

00:04:33 There will be a new version of the BEx Query Designer and this new version will—as all the new tools which we have shown in this course—

00:04:41 also be contained in the Eclipse environment.

00:04:44 Right. Also there we are bringing some new functionalities for the modeling experience. But now let's really take a look at how this works in detail

00:04:54 because this is something a lot of people forget or underestimate, what actually the important role of the OLAP manager is here in this case.

00:05:04 So here we see a fairly small example of an analytic query result

00:05:09 with two characteristics we have here in drilldown and three key figures in the right part.

00:05:16 The first key figure is maybe the simplest one. That's just a quantity of products ordered in a certain country. So for pencils we have 10,

00:05:23 for paper we have 5, and we have subtotals here that just sum up the numbers.

00:05:27 And here we have another subtotal and we have a grand total.

00:05:31 It's a summation, so no big complex...

00:05:35 But we will see that even this simple or innocent-looking query result has some tricks about it.

00:05:40 So for example, when we go to the next row, the number of customers gets more tricky already

00:05:45 because if you look at the individual numbers, the 5 and the 3, they are pretty easy to

understand and you can just believe them,

00:05:53 but if you look at the subtotal here, it's maybe a little bit surprising at first because 6 is not the sum of 5 and 3.

00:06:00 Obviously what happens here is this means we have 5 customers buying pencils and 3 customers buying paper, but we must have two customers buying both.

00:06:10 That's something which we cannot detect from this result because we don't see the customer in the result here.

00:06:17 So what we really would like to calculate there are the distinct customers. So how many distinct customers, no matter what they are buying, exactly, do we have?

00:06:26 And this is expressed here in this definition.

00:06:29 The other tricky part about this query is if you look at the third key figure here, the share per product,

00:06:35 it's also something where you need the result of a subtotal in order to calculate the lowest granularity within the query result.

00:06:45 So that actually shows you that in order to calculate the whole query result,

00:06:49 you have to do certain subtotal calculations before you can actually calculate the value of cells at the lower granularity.

00:06:59 So this means there is a huge dependency actually between the different key figures because you need the first one to calculate the second one.

00:07:06 And this important orchestration role is done by the OLAP engine. And you just have to imagine how would it look if you do it by your own SQL way.

00:07:16 Yeah, if you want to especially count the customers in a correct way and detect that you have customers buying both products,

00:07:22 then you'll basically have to write a SQL statement looking like this where you take the customer into the drilldown, basically, and retrieve a result set like this.

00:07:33 Now, what you see immediately is that this result set is larger than the result set which you see as the total result of the report.

00:07:40 And even worse, if you think of the combinatorics behind the problem, if you have a lot of materials and a lot of customers

00:07:46 then this result set can really explode. Think of thousands of customers, thousands of materials, then this is potentially a huge result set.

00:07:55 Now the problem in this case is even if you have a database which can process this result set in a very fast way,

00:08:01 you would still have to transfer it to an OLAP tool on top of the database. So if you have an OLAP tool installed on your front end or, as in the BW case, on an application server,

00:08:11 then the data transfer between the database and the application server or the desktop would be very, very painful.

00:08:19 And that's the whole point about the pushdown idea in BW. That's why we want to push down more than SQL processing down to HANA.

00:08:29 So in the very end, it's good that you have an engine there which also does some intermediate results to do some temporary

00:08:38 automated calculation without that you have to bother with it. And this is really already, what I said already, the orchestration role of the BW Analytic Manager in this case.

00:08:49 So what we can learn from this example is that you basically have to analyze the final structure of the report

00:08:59 and see what order you have to calculate certain intermediate results. So you basically can derive from the layout of the report, which you designed as an end user in the Query Designer,

00:09:10 the BW Analytic Manager actually derives the so-called OLAP graph which orchestrates the required subcalculations in the right way.

00:09:20 And that's what you see here. It looks pretty complex and we're not going to go into it in detail.

00:09:24 But for example, you see that in order to calculate...I think we saw this level L2 here, which is the share, we first need the level L1.

00:09:35 That's why the L2 has to be calculated after L1.

00:09:38 First need the quantity to calculate the share.

00:09:40 So that's indicated in this graph here, that L1 is basically a prerequisite to calculating L2. And if you think about the problem in detail, you will see that L3 and L4 also need the result of L1.

00:09:51 So this is basically how it looks behind the scenes, but also an interesting point is really how does it look on database level?

00:09:59 Well, the important thing is that you don't have to bother about this because the BW Analytic Manager does all this analysis of the report which you just created in a graphical way

00:10:09 and creates the right orchestration out of this and all the intermediate stuff and you don't even have to know about it.

00:10:17 If you compare now if complex things like an exception aggregation are really being processed by the Analytic Manager,

00:10:25 there is a huge difference between BW running on any DB or BW powered by SAP HANA.

00:10:32 And this is really again going into the whole pushdown direction.

00:10:38 So one of the fundamental differences is that when BW, the BW application server, communicates with any DB, then it uses plain SQL as an interface.

00:10:48 And with SQL, we've seen that we cannot push down all this complexity because some of these layers, some of these subcalculations which we just described,

00:10:58 cannot be easily expressed in SQL.

00:11:01 So therefore we typically have to retrieve a larger result set like the one we saw in the second slide where we had this SQL example.

00:11:10 We have to fetch this from the database into the application server, do the additional calculation on top of this, and then can present the result to the front end.

00:11:19 Take again our sales order example, what we just demonstrated in the Query Designer. We first, in this case, would have to bring up all the different sales orders,

00:11:27 store it in the application server in an internal table, and then we would count the different distinct values.

00:11:33 With SAP HANA as database, this is happening again totally different. So we stay at database level,

00:11:40 we generate the so-called calculation scenario for this specific query within the database,

00:11:46 and then we are just bringing up the final result to the application server and can give it out of BW to the reporting tool.

00:11:54 And the result is basically that even if you're processing a huge amount of data in the database, the transfer to the application server on the front end will be much, much smaller.

00:12:04 So think of the initial example which we had with a fairly small report and think of maybe 1,000 products and 10,000 customers.

00:12:12 Then you will still have a fairly small query end result for the front end, but a huge intermediate result which, in this case, can be totally processed down in HANA

00:12:22 and therefore avoids a lot of network traffic compared to the classical way of communicating via SQL.

00:12:29 Okay. So what is being pushed down already? Actually, these are really just a few examples. We don't want to cover them now in detail.

00:12:39 But you will find, in the handout material, an SAP Note, a list containing what are the specific functions

00:12:47 which we from SAP side already optimized for SAP HANA usage.

00:12:51 You can imagine we have so many different OLAP functionalities, so many combinations.

00:12:57 You can create for example, an exception aggregation in the Query Designer

00:13:01 and you will find there the detail of what combination in which case is being pushed down. And maybe for certain functionalities, what are we planning there.

00:13:11 Okay, so let's sum it up. The OLAP engine

00:13:16 is really a unique and a strong set of functionality to do analytic functionalities.

00:13:24 It's really an engine, a transparent way to create query results without that you have to manually write SQL statements.

00:13:35 And, at the very end, nearly all OLAP functionalities—you have the list there—are getting pushed down to SAP HANA.

00:13:42 And in the next unit, we will see a lot of examples of those.

WEEK 3, UNIT 5

00:00:13 Hello and welcome back to week 3, unit 5: Pushdown of OLAP Functions to SAP HANA.

00:00:19 In the previous unit, we've seen the capabilities of the Analytic Manager, also called the OLAP engine, in theory.

00:00:27 And in this unit, I think it's now time that we show them in action. So what we will do in this unit is exclusively in the system demonstration, so you will see

00:00:38 ome OLAP features in action. We will experience during the reporting the performance of SAP HANA.

00:00:45 And we will use the SAP Design Studio application to build our own reporting application.

00:00:53 So what we are doing there is we've basically set up a Design Studio application and embedded there all the

00:01:02 BW queries we've built based on our EPM model.

00:01:08 The whole scenario is again based on the CompositeProvider which we showed you earlier containing the sales order details.

00:01:14 o here is Design Studio, where you create such an analytic application. Now let's have a look at the result of this work.

00:01:22 So here you see the analytic application which we are going through in detail now.

00:01:27 On the left-hand side, top left corner, you see certain ad-hoc analysis items.

00:01:32 So that's basically a number of BW queries which you can open successively to get detailed information.

00:01:39 And in the background, BW queries based on this CompositeProvider will be executed.

00:01:46 You see also—and we will cover that of course—that we have provided several dashboards because this is obviously also a very nice way to consume the data of SAP BW.

00:01:56 So we have a dashboard for our KPIs and we have a dashboard just to demonstrate that

00:02:01 reporting or consuming data from SAP BW can even be consumed by any mobile application or device and this is what we are going to demonstrate.

00:02:10 And the last step will be to show you here, in the lower part, a comparison between HANA and non-HANA execution of the same query

00:02:18 to really show you the performance benefits which you gain with HANA.

00:02:21 Now I guess let's start the ad-hoc analysis part. The first query which we want to have a look at in detail is the EPM Vendor Information query.

00:02:31 So in background, as we said, we are now consuming the data out of our main CompositeProvider,

00:02:38 joining the information of the open and completed sales orders, and overall we have there a data volume at present around 14 million data records.

00:02:49 Maybe a few words about the general layout of this page here.

00:02:53 So that's one of the preconfigured templates which comes with Design Studio.

00:02:58 It basically contains some basic functionality which is very useful in such a report.

00:03:06 So you can, for example, switch from a table layout to a chart layout. You can adjust filters. You can also change the drilldown state. Maybe let's open this.

00:03:16 So here, according to what you specified as free characteristics in the query, we have the possibility to change the drilldown state.

00:03:24 And another thing you can do here is, for example, use hierarchies.

00:03:31 If any of the characteristics in this report contain a hierarchy—we'll later see an example of the product—

00:03:39 then you could obviously start using this hierarchy and then change the drilldown accordingly.

00:03:45 We could also change the scaling factor, so how many zeros we are seeing for the key figures.

00:03:51 So this is really out of the box so it was not hard for us to create this template or to use this template. It was quite easy

00:04:00 and therefore we have now this nice ad-hoc analysis area with all the BW queries.

00:04:05 Now let's have a look at the query result in detail. So we have a drilldown by vendor and order currency.

00:04:11 The key figures we're looking at are the sales order value, the net sales order value—which is one of the key figures of the CompositeProvider—

00:04:17 and we have two other key figures which are calculated on top of this data.

00:04:21 So we have the number of sales orders, which is obviously not a part of the CompositeProvider but it can be and has to be calculated from the data in the CompositeProvider,

00:04:31 and once you have the number of sales orders for each vendor and order currency, you can also compute the average value in any given currency.

00:04:42 Now we'll show you how to do this in the Query Designer. So we'll have a look at the back-end query here.

00:04:50 And you remember the orchestration functionality of the Analytic Manager,

00:04:56 that one key figure is depending on the other and we know exactly in which order we have to calculate. This is exactly what is happening with the average number of all sales orders here.

00:05:07 So here you see the three key figures. The number of sales orders is, in BW language, an exception aggregation.

00:05:15 If you look at this here, it basically says count all detailed values with respect to sales order.

00:05:21 So we count all sales order values here, all the different sales order numbers here,

00:05:28 and then in the definition of the sales order, the average sales order value, we take the sum of the sales order value and divide it by the number of sales orders.

00:05:38 So that's exactly what you mentioned. You first have to calculate this number here, which is already an exception aggregation on top of the data which is contained in the CompositeProvider

00:05:49 and then use this result as a prerequisite to compute the average.

00:05:54 Also for a recap purpose, how would this be executed on any DB? It would not be that efficient.

00:06:03 I saw a lot of customer implementations where exactly this exception aggregation counting is really, with a large data volume, a painful operation

00:06:13 because it's happening in the application server. And in our case, we are doing it directly within SAP HANA because this is optimized with the usage of SAP HANA.

00:06:22 Okay. Maybe let's come to the next example here. That's a totally different type of query.

00:06:29 It's called Price Trend Analysis, and maybe we're not going to go into it in too much detail,

00:06:36 but it basically show you—and that's also very nice to see—a historic drilldown state.

00:06:41 So you see how the average price of a certain product changes over time.

00:06:49 So here we have calendar year and month and you see how the product price on average changes. You also see the average sales order quantity with respect to this product and how it changes over time.

00:06:59 Especially when you're interested in time series and this kind of stuff, then this is a very interesting thing.

00:07:08 The last and maybe most tricky query and most interesting query is the Top 10 Products of a Customer by Sales Order Value.

00:07:17 So let's have a look at this one. We will also show you how you define such a query in the Query Designer and what the necessary components are.

00:07:24 So this is now combining a lot of OLAP functionalities in one single report. Obviously we are selecting here not every product. We are only selecting a top 10 of our products

00:07:34 based on the net order amount we are doing with it.

00:07:39 So let's see what this looks like in the back end.

00:07:45 And you'll probably also see why it's actually the top 15 and not the top 10 products.

00:07:52 If you're curious about that, you should open the query in the Query Designer which, by the way, you can do in your cloud system if you have one.

00:08:04 Which we recommend, because they you can play around with that as well.

00:08:08 So here we have on top of the key figures which are defined here, as we've seen this, the drilldown state product ID,

00:08:15 and we have the net order amount. Those were things we've seen before.

00:08:20 Now let's have a look here...it says the description is top 10, but actually we changed this to top 15.

00:08:29 So maybe let's see what happens if you do an online change here.

00:08:33 So I really changed the query definition now. Save this.

00:08:38 And let's just reload this same query. The interesting thing is...

00:08:44 It changed.

00:08:45 It has changed.

00:08:47 And the reason for that is that the queries of BW are embedded in Design Studio as so seen as a data source with the connection parameters and so on.

00:08:56 but a definition obviously is still managed by BW. So if we change something there, save it,

00:09:02 it will be directly popping up here if nothing there is overruling, because in Design Studio you could also define it by heart, but we are working here dynamically.

00:09:11 So top n was one interesting aspect about this query. Another thing which we wanted to show is hierarchies.

00:09:19 I think there is even...it's the first time we're showing you everything based in euro.

00:09:27 And the reason for that is we are doing a currency conversion. So you have seen that we have many different currencies available in our system,

00:09:36 but in this specific query we have assigned a conversion type for the currency

00:09:43 and there you see we are now looking up in the back end, during runtime, how the currency for euro was and then we are

00:09:53 calculating everything accordingly during the runtime.

00:09:56 So the definition of this currency translation type specifies all the details like what date is relevant for doing the conversion.

00:10:06 All this kind of stuff is maintained in this currency translation type, 0EPM_DEF. You can actually check this in the ABAP back end to understand what's really going on.

00:10:17 But what you see is we do a currency translation from the currencies contained in the underlying InfoProvicer to European euro.

00:10:28 That's why the results are all shown here in euro, which is of course very important because otherwise you would sort according to different currencies, which doesn't make sense.

00:10:38 And then hierarchies: the feature which we also wanted to show.

00:10:44 The product is obviously a characteristic which comes very naturally with a hierarchy.

00:10:54 So let's activate this hierarchy and see what happens.

00:10:59 You actually then can drill down by certain product categories until you come to the lowest granularity.

00:11:06 So you see the different product categories which we have in this company. That's computers, keyboards, PCs, whatever you see here.

00:11:15 This hierarchy is an object, a reporting element, of course coming from the back end, so this is related to the InfoObject of the product

00:11:25 where this is an important master data piece of the whole InfoObject service. And then we can reuse this in the front end.

00:11:34 We have loaded it in the back end and there you can go and switch it on here in Design Studio.

00:11:39 We could actually even switch between different hierarchies if we had one. We could have time-dependent hierarchies so all this is possible.

00:11:45 From a HANA perspective, even parts of the hierarchy calculations like intermediate results if

the hierarchy gets more complex

00:11:53 are also aligned with SAP HANA and therefore also we have here an optimization due to SAP HANA.

00:12:00 Okay, I guess that's it for the ad-hoc part of the queries.

00:12:04 I think it's time to do a bit of dashboarding, because this is something which is very trendy and of course our managers are really liking

00:12:14 the view on the dashboards where you have different KPIs directly popping in your eye.

00:12:20 And what we also have here is we are calculating now every sales order we have in the back end. You see the amazingly large sales order net value behind that.

00:12:29 And then even we are doing again exception aggregation. We are taking the average of each single sales order.

00:12:38 And even such a dashboard can have some interactivity built into it,

00:12:43 like here we could, for example, switch to a different currency. Let's try that.

00:12:50 It's easily possible. Yeah, let's see. So now we have selected...let's try to get rid of euro here, and apply this, and then you see the whole thing in U.S. dollars.

00:13:01 Also there are a lot of key figures which are again exception aggregation and the speed for all the sales orders is quite efficient.

00:13:11 We are making far more business in euro, by the way.

00:13:14 That's right. That's exactly what you would like to find out with a dashboard.

00:13:19 But this kind of view, it's the one thing, but if you go to the next dashboard,

00:13:24 we will also see that even for mobile applications, for mobile devices—like this is now optimized for an iPad, for instance—

00:13:33 we could have and achieve a reporting based on SAP BW.

00:13:37 So this, I mean running this on a tablet would probably be much more interesting, but you see that you basically have the...

00:13:44 even here on the desktop, the typical feeling of a mobile application. So you can swipe back and forth.

00:13:53 But you still have the possibility to go into detail. You can restrict certain characteristics

00:13:59 like product and business partner and really do a detailed analysis even with this rather simple...

00:14:06 Right. But this is...I mean, at latest, at this point in time, you really need the speed and a stable query runtime for everything you are doing.

00:14:16 And again, this is the same logic, so it's just a different way how we display the data. We have here again top n

00:14:24 top 10, top 5, of our customers, with the same set of exception aggregations behind.

00:14:32 And it's still working and feeling on a very good and stable manner here.

00:14:38 Okay. Now the final thing we wanted to show you is the comparison between HANA and non-

HANA.

00:14:44 So it's basically what's going to happen when we open this is we will open the same query in two ways. One is running in the HANA-optimized way and one is running without any HANA optimization.

00:14:55 Of course, both queries are running on the same system and you will see the difference in speed immediately.

00:15:03 Unfortunately, we have a very small screen resolution here, but what you see already is the first query result with HANA optimization is there already

00:15:12 while the other one...maybe I'm closing this.

00:15:16 Where it's still loading.

00:15:17 Yeah, it's still loading. This is probably going to continue for...

00:15:20 Three minutes? I don't think that we have the time. But you'll see, when you try it out, that the result is the same.

00:15:29 What we just did to simulate this a little bit is we switched off, we copied the query two times, and if you go to the properties for the

00:15:39 non-HANA execution query, you see that here this is an interesting flag where you can define where and how the query result—

00:15:47 especially the exception aggregation or the counting in the calculation—where this is going to be operated and here we switch it simply off.

00:15:54 So we say no optimization, no operations down to SAP HANA. So this is how it traditionally would be executed in any relational database

00:16:03 where no BWA or SAP HANA would be available and then you have the same query.

00:16:09 From a runtime perspective, I think we have around 4 to 6 seconds, something like that, for SAP HANA and

00:16:17 the other one, I think it's around 3 minutes, but you can try it on your own and see that it's not faked.

00:16:24 Okay. So I guess that's it for the demo. Let's wrap up what we learned.

00:16:30 So as in the previous session, you've seen that the BW Analytic Manager really we have a huge variety of

00:16:37 predefined functions and methods to do and to achieve a very nice reporting there,

00:16:44 especially when it comes to currency, to unit conversion, to exception aggregations.

00:16:49 This is where really we have a key strength here of the Analytic Manager.

00:16:53 Yeah. And we have also seen that you can actually use very fancy front ends and very modern front ends.

00:17:00 We've even shown you basically a mobile experience based on a BW query. So the BW query is really a very

00:17:08 good and powerful layer to model business logic as the end user wants to view it,

00:17:14 and then connect this to several possibly different front-end tools.

00:17:22 So I think it was enough now for the BW OLAP capabilities. In the next unit we will focus on another very interesting SAP HANA feature,

00:17:31 which is the SAP HANA analysis process.

00:17:33 And we hope you are well prepared for the self-test now.

Week 3, Unit 6

00:00:00 Hello and welcome back to Week 3, Unit 6, SAP HANA Analysis Process.

00:00:18 In this unit we are going to cover a brand new feature with SAP BW powered by SAP HANA and especially with our 7.4 release.

00:00:28 We have a new Analysis process available which is HANA-optimized and we are now going to cover the different components and then

00:00:37 give a system demonstration, which is a starting point for a business scenario which we will also later cover in our Planning unit.

00:00:44 So, here is a rough overview of the idea of the HANA Analysis process. On the right hand side you see what you can basically do with a HANA Analysis process;

00:00:54 that is, take the data of a BW InfoProvider, run some, maybe very complex analysis on it, some fairly advanced algorithms, and then you can write the result of this algorithm back into another BW info product.

00:01:07 The good part here is that this process is being natively optimized and exclusively there for SAP HANA usage,

00:01:17 which means again we are doing a certain pushdown of the functionality. This means we take the information out of, you see there, two structures,

00:01:27 two lets say InfoProviders, we will cover that in detail, push down the calculation through SAP HANA with certain functionalities which are exclusively coming from SAP HANA,

00:01:37 like you see we are clustering association regression analysis; and then write the result into a certain InfoProvider.

00:01:47 The whole logic is managed by the BW process.

00:01:51 Management means we are fully integrated in the process chain, scheduling the data transfer process and so on.

00:01:59 Since you mentioned data transfer processes, at first sight this picture looks a little bit like what you did in transformations, but from the idea, it is actually a little bit different,

00:02:10 and the difference is basically that these algorithms here are pretty advanced. And typically, they don't work record-by-record,

00:02:17 as you would operate in typical transformations, but they operate really on the whole dataset. So you really have to analyze the whole dataset, or at least big chunks of the dataset,

00:02:28 maybe do certain sub-aggregations, all this kind of stuff. So you really start with the whole dataset as an input and analyze it,

00:02:36 run the algorithm on it, and the result is then again written into another InfoProvider.

00:02:42 From a scheduling perspective, it's very similar to transformations because you typically do this also on a regular basis,

00:02:49 daily, monthly, whatever; but the type of algorithm is different from what you do in a transformation normally. You might also think that this is similar to the analysis process designer,

00:02:58 which was our tool to do some similar things in the data mining area, and there might be some

certain overlap in terms of functionalities,

00:03:09 but this is really our feature also for the future, which is optimized for SAP HANA and offering also

00:03:16 a few functions which are not possible with the APD. So let's come to the details here.

00:03:24 The three parts which the modeling of an HAP normally consists of: You have to specify a source, right. The source can be BW InfoProviders

00:03:33 and database tables, and you see the details here on the right hand side in the SAPGUI popup.

00:03:39 Right and this is basically covering everything we know in BW; it could basically be every InfoProvider;

00:03:47 it could be MultiProviders, that means virtual objects could be persisted; it could be InfoObjects, so really the full range of our objects,

00:03:56 and even database tables which are sitting directly within SAP HANA. Then the type of functions which you can execute: There are a lot of predefined functions.

00:04:05 A lot of libraries which natively come with HANA or are additional components that you can install on HANA, like the Advanced Function Library, predictive functions, that kind of stuff.

00:04:16 But it is also possible to write your own algorithms, obviously, using L, R, SQLScript, all that is possible.

00:04:23 From a target perspective — so that's the third component — it's basically that we can write into a so-called analytic index,

00:04:32 which is a structure which we did not mention so much, so far,

00:04:38 but let's say this is a kind of temporary structure that you can use to persist data in certain operations.

00:04:48 Then obviously DataStore objects, which as you have seen play an even more prominent role in the BW with 7.4 SP8.

00:04:56 It could be a database table again, coming from SAP HANA, or it could be a so-called stacked scenario, which means your input for a HANA Analysis process

00:05:08 is another HANA Analysis process, and the result is then your input to start the calculation of the following process.

00:05:17 Okay, so I guess it's demo time. So what we are doing there is we will start to prepare a planning scenario.

00:05:26 And we are doing that based on our — you see that here — on our scenario for sales orders. So we will take the input information of our opened and completed sales orders.

00:05:37 We will then calculate a cluster information for our business partners.

00:05:43 So it means what is our top group of business partners we made the most revenue with.

00:05:51 And then later, in the next unit, we will then start based on these cluster functions, clustering groups. We will then start a planning scenario.

00:06:00 So let's say I would like to do next year 10% more money with my group A, or with my group B, and this is now the starting point.

00:06:08 So we take our standard InfoProvider, which was the CompositeProvider, it's this one, right. And we saw in previous demonstrations

00:06:17 that we have around, I think, 8 million records there. And this is our input for the HANA Analysis process.

00:06:26 So, let's go into details here. Again, on the first step you see an overview of the process, which actually contains exactly the information which you just mentioned: The 3 major components,

00:06:37 the data source, CompositeProvider which you just mentioned; then the function which we are going to operate, which we are going to execute, which we want to use,

00:06:45 and in this case that's a K-Means algorithm, which basically does a clustering of your data according to K clusters.

00:06:53 So, you're given the dataset, let's say you have the customer and revenue associate to a given customer and you want to group customers into K groups,

00:07:03 which kind of form one individual cluster each.

00:07:09 That's what K-Means, or is therefore, and you can see also a number of other algorithms which are possible here.

00:07:18 So another way of clustering would be an ABC analysis that works a little bit different; they are different, they prescribe certain percentages,

00:07:26 how you want to group customers. K-Means works a little bit differently, you just specify the number of clusters. And you see that we have some additional algorithms

00:07:36 here as well. So it's basically something like an ABC analysis from the idea. So this will be our target structure,

00:07:47 but with more details and more complex and also more, as an output of the data, data mining functionality.

00:07:55 And we will also provide in the handout, more information about that, that you can really see what is happening there.

00:08:02 And there is a data target. Maybe you have seen this already in the RSA1 structure which we just opened. The whole thing is embedded in a data transfer process here,

00:08:11 so we are not writing the results directly into a data target via the HAP, but we actually use transformation to map the results of the

00:08:21 function here to a data target in BW, and we use the DTP for scheduling.

00:08:26 Now here we specify details about the data source. On the right hand side, you see the input structure of the algorithms.

00:08:35 So K-Means has one input parameter, which is the Object ID. Object ID in our case is the business partner.

00:08:42 So because this is up with your object you would like to do clustering. So this means we take the business partner, we take the business partner role again

00:08:52 because it's like we saw in the previous demonstration, it's a compounded object, and then the next step would be to say hey, this is my object,

00:08:59 which is my input for my process, and then I need, of course, key figures which are important then to do the reference

00:09:09 and the calculation. And this is the net order amount, and you know also that we have different currencies per business partner,

00:09:16 because we are an international company, and therefore you see also the currency key here. It means we can even do a currency conversion in the background

00:09:26 to do then the calculation of what cluster does a business partner belong to.

00:09:34 Now in the Data Analysis tab, you specify the parameters which the algorithm needs in addition. Here, we have a parameter Number of Groups;

00:09:43 that's basically the number of clusters. So we are saying we have a huge number of customers.

00:09:47 We want to cluster them in 3 groups. The other parameters are maybe not that important. Maybe this one is important,

00:09:54 K-Means is an iterative algorithm, so we basically specify how many iterations we want to run.

00:09:59 We say 100 iterations should be fine for the accuracy which we want to obtain.

00:10:04 The other parameters if you don't know about them, it's actually very nicely described in the help documentation of SAP HANA,

00:10:15 because these functions are basically part of HANA libraries; therefore you can actually find all of the details in the HANA documentation.

00:10:24 When it comes to the data target, we see the output structure here. That's what we also... no we did not see it;

00:10:32 you see the business partner again in the output structure. We see the computed cluster ID which we basically want to derive, and then there is the distance,

00:10:42 which basically describes for each business partner the distance to the center of the cluster.

00:10:48 You will see that in the handout information what the logic is behind, but this you will see, this will behave like, a bit like a key figure later,

00:10:56 which could then be an input and be a field for more complex calculations.

00:11:02 We will try, and we would like to give you now just a purpose of what you can do, and therefore we keep it simple at that point and I think we could,

00:11:10 maybe before we run it, let's have a look at the transformation.

00:11:15 So here we see what we just saw, the output structure of the algorithm, and we use the transformation basically only to map the output structure of the algorithm

00:11:25 or of the HAP, to the fields of a DataStore, or the InfoObjects of the data store.

00:11:30 This is the latest object that we created beforehand with these 4 fields, and this is now being our target structure again, and it could also be an analytical index,

00:11:40 it could be a database table. We are using a DataStore object because later we would also like to do a reporting on top of it, so you will see

00:11:51 that we will bring the sales order information back together with the cluster information. Ok, so for scheduling we said we are going to use the data transfer process,

00:12:02 which we have also prepared for you here. You see also that this is again HANA-optimized, so

the execution will be happening down in SAP HANA.

00:12:12 So let's run it, jump into the monitor. It's quite fast, I think it's actually done already.

00:12:18 Remember this was operating on about 7 or 8 million individual line items, right. We have 7 or 8 million sales orders

00:12:28 in the background and the result is a number of 46 records, which basically means we have 46 distinct customers I guess.

00:12:36 And let's see what the result of the HAP looks like. Here we see... So there you see the

00:12:47 information of which business partner belongs to which clustering and you see, okay,

00:12:58 now I have my cluster information on the left hand side, clusters 0, 1, and 2. On the left hand side we have the business partner together with it's role and the distance, that's what I just said,

00:13:08 that basically is an indication of how big the cluster is, right. So this is nothing you would like to use of course for reporting,

00:13:15 therefore we use this structure here just to persist it. And if you now combine this information again with the sales order

00:13:22 and you do an inner join, this means that you would only like to have those business partners where I do have cluster information.

00:13:32 And let's go to Analysis for Microsoft Excel and we select the CompositeProvider. We have also already shown in the CompositeProvider unit,

00:13:42 this was our inner join scenario where we did a join exactly out of 2 DataStore objects — this one here — which is the result of our cluster analysis and the sales order DataStore object.

00:13:56 So let's go right into the CompositeProvider again, which contains the join, that's called YEPM_CP_02,

00:14:06 M is missing, not our name space here, YEPM_CP_02, there you go.

00:14:13 Right, and then you see that the join information. And maybe just use the cluster. So now you basically see information

00:14:24 by cluster. And now we can for example drill down by order currency to get more information. So we actually see the amounts,

00:14:34 the drill down by currency for each of the clusters. If we now also take the business partner into account,

00:14:43 you see the input scenario for our planning scenario, so you see cluster 1 contains the following info.

00:14:50 The following business partners with a certain revenue or net order amount and different currencies, and this is our situation to start then later to do planning, to say hey,

00:15:01 I would like to do 10 persons more in cluster 1, or maybe even with my customer Becker Berlin,

00:15:08 I would like to do maybe even a bit more, and I can manually do the planning step here.

00:15:14 Okay, I guess that's it for the demo and it's time for the wrap-up.

00:15:19 So the SAP HANA Analysis process is a great opportunity within the latest release of SAP BW

00:15:28 in combination with SAP HANA to implement predictive analysis to do more enhanced data mining scenarios

00:15:38 directly within SAP BW on InfoProvider data. We have different functions,

00:15:45 different libraries available, and the data processing is exclusively happening within SAP HANA.

00:15:51 That's it for this week. We will have a weekly assignment for you and hope to see you again next week.

www.sap.com