WEB SERVICES FOR ENTERPRISE ERP

N. Vinoth, S. Sterose

ABSTRACT

The objective of this paper is to clarify the ideas, info and steps to make a chic hybrid application that may perform transactional and analytical operations in real time, improved performance, simplified design by victimization based mostly internet service and supply a responsive user friendly interface for this ERP application. This cannot be achieved by one technology, however by the convergence of the new technologies mentioned within the paper.

One of the core elements of SAP HANA XS is to produce http(s) access to the underlying knowledge among SAP HANA by investment open technologies like OData, REST, JavaScript and HTML5. We'll discuss examples on a way to expeditiously expose the Hana knowledge read victimization open standards like REST, OData, JSON and to render the result among a HTML computer program victimization SAPUI5

***Keywords**:SAP HANA,OData,SAPUI5, SAP HANA XS,JSON, CO-PA*

INTRODUCTION

As a general rule sculptured views (i.e. Calculation, Analytical and Attribute views) ought to be thought of as a primary alternative once developing content for SAP HANA[4]. they're style time objects and once deployed as runtime objects they become extremely optimized in-memory[1] columnar information views that may exploit the assorted optimizers among SAP HANA[1][3] with the only real intent tomax performance. Since the business logic is embedded directly among the models this mechanically permits calculations and filters to be pushed down and parallelized deep among the information. What distinguishes sculptured views from SQL views is that they generate a dynamic execution path that supports question remotion and be part of omission and can, for instance, solely perform the calculations against those columns requested creating them smart candidates for big knowledge sets wherever best performance is needed. it's necessary to think about the subsequent basic principles once developing content for SAP HANA: initial, make sure that business logic and calculations square measure pushed down into the information either via sculptured views or procedures. Second, minimize the info movement between completely different layers. To totally capture the performance advantages of SAP HANA adequate attention must tend once planning applications so as to make sure exploitation of the underlying engines supported the applying necessities. The CO-PA state of affairs in our examples utilizes many engines that square measure hid from end-users. Initial the heavy-lifting computing sales-data is retrieved and processed victimization many Analytical views dead by the OLAP engine. The collective reduced

results-sets from these reads square measure encapsulated and combined among a Calculation read dead by the Calculation engine. Successively the results of the Calculation browse unit of measurement two-handed over to the enginethe data-set is then sent to the userlogging via the intrinsic SAP HANA XS web server [4].

RELATED WORK

We will discuss additional a demand on a way to expeditiously expose the Hana knowledge read victimization open standards like REST, OData, JSON and to render the result among a HTML computer program victimization SAPUI5[3].

II.1. Calculation view

The baseline Calculation read that's to be exposed through http(s) implements the classical union with constants pattern that mixes completely different input sources within the best fashion for in-memory columnar process. the info is classified by Sales Organization and Material and comprises many calculations that may eventually be employed by the business to produce insight into product profitableness.

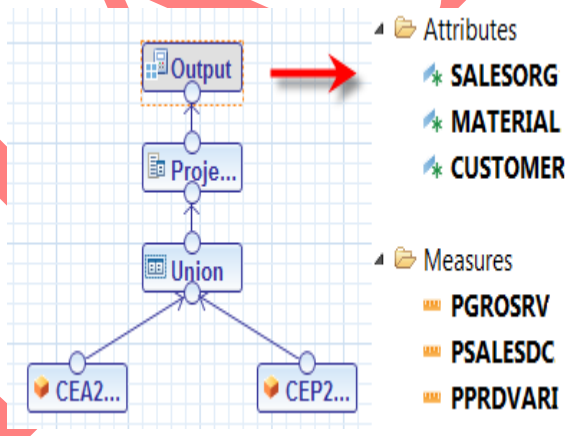


Fig.1. Graphical Calculation read

In our example, the subsequent SQL statement and result-set satisfy the business state of affairs bimanual to America as a demand for our development project among SAP HANA XS.

SALESORG	MATERIAL	PNETREVN	ANETREVN	PGROSRV	AGROSREV
1000	DPC1002	6744345	6254745	6945511.4	6456471.72
1000	DPC1003	7419024	6108677	7647158.74	6305714.09
1000	DPC1004	7946067	5683390	8174822.31	5861671.93
1000	DPC1005	7841713	7528246	8086496.96	7765368.06
1000	DPC1009	939636	1357671	969849.64	1401447.53
1000	DPC1010	945541	1293828	972540.18	1335257.7

Fig.2.SQL results

II.2. XS Project

In order to develop and deploy SAP HANA XS content the various artifacts (i.e. text files) should exist initial among specific native comes. Throughout this paper we'll be victimization the standard SAP HANA Development XS Project. This section consider that the reader has some familiarity victimization the SAP HANA Development Perspective [4] and is ready to make project and link the local project to the repository among SAP HANA for ASCII text file and version management.

XS comes contains many completely different development artifacts looking on the wants of the applying. needed files that enforce authorization have extensions (.xsaccess), (.xsapp), and (.xsprivileges), ex gratia files embrace server-side Java Script (.xsjs), and client-side UIs victimization SAPUI5 (.html). OData Service Definition Language files have extensions (.xsodata). in contrast to the SAP HANA Studio creatorperspective that have refined UIs for making models, content for the SAP HANA XS may be created victimization easy text editors that return customary among SAP HANA Studio. it's necessary to notice that XS comes additionally contain specific libraries called the SAP HANA XS JavaScript genus Apis that ought to be used once making server-side JavaScript content.

Figure three below displays a typical XS Project structure; the CO-PA example artifacts square measure hold on in a very native project referred to as (copaproj) and connected to a repository package referred to as (copaxs) among SAP HANA.

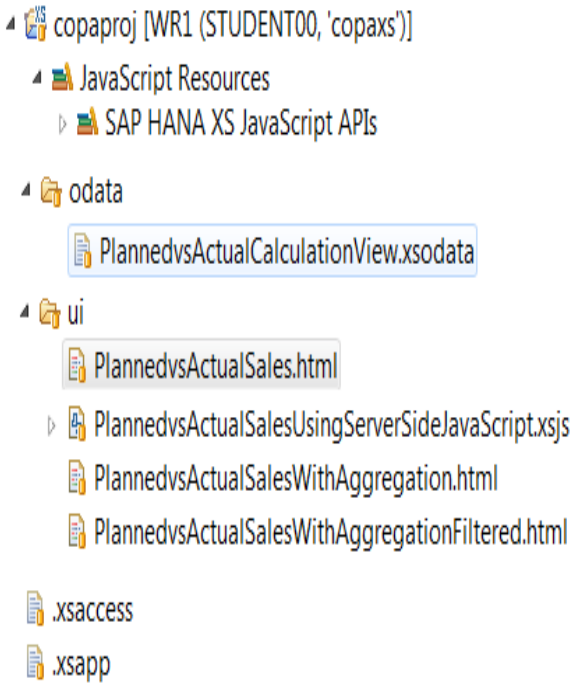


Fig.3. XS Project example

Depending on the http(s) port of SAP HANA, any hypertext mark-up language pages, server-side JavaScript pages or OData services is also served by victimization address syntax like:

Similarly it Definition Language files is also documented an identical suggests that [3].

II.3. OData

The first steps once exposing content among SAP HANA through OData is to make associate OData file referencing the actual information object. though Calculation views square measure employed in the instance a similar principles applies to Analytical views[4].

```
1 service {
2   "copa::PLANNED_ACTUAL_SALES_CV" as "PlannedvsActualCalcView"
3   keys generate local "ID"
4   aggregates always;
5 }
```

Line 2: The <::> notation specifies the Calculation hold a runtime object that's situated among the package. it's additionally attainable to reference style time objects. For details confer with the SAP HANA Developer guide[4].

Line 3: Objects that don't have a novel key in their results (i.e. Calculation views & Analytical views) needs the generated key entry. bear in mind that this key price numbers the result beginning with one and isn't meant for dereferencing any item within the result set. The key's valid just for the

length of this session and is employed solely to satisfy OData's demand for a novel id within the results[3].

Line 4: is needed for Calculation views and Analytical views wherever the knowledge concerning attributes and measures square measure hold on among specific data tables. which means that the combinationperpetually statement can make sure that this info springs from metadata; thence the notion of derived aggregation. This but means whenever the underlying Calculation read changes this OData Service Definition Language file must be re-activated. this may be overcome in line two by referencing a style time object rather than a runtime object.

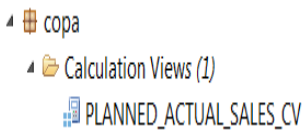


Fig.5. Package structure using Navigator view

Once it activated then the file could also be remarked as using a program as follows.

```
<service xml:base="http://ld9506.wdf.sap.corp:8000/copaxs/od
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://w
  <workspace>
    <atom:title>Default</atom:title>
    <collection href="PlannedvsActualCalcView">
      <atom:title>PlannedvsActualCalcView</atom:title>
    </collection>
  </workspace>
</service>
```

That a group may be obtained by appending to the address.

Hint: bear in mind of the subsequent meaning that applies to multi-dimensional reportage (i.e. Calculation views and Analytical views) and specifically once aggregating knowledge. so as to combination, the <$select> parameterought to be used because it can inform the OData service to come back associate collective set.

```
<entry>
   <id>/PlannedvsActualCalcView('1')</id>
   <title type="text"/>
   <author>
     <name/>
   </author>
   <content type="application/xml">
   <m:properties>
    <d:SALESORG m:type="Edm.String">1000</d:SALESORG>
    <d:ANETREVN m:type="Edm.Decimal">379091384</d:ANETREVN>
   </m:properties>
</content>
</entry>
```

Fig. 7. OData service response

At this time we've got smart understanding a way to construct the address question so as to decision the Calculation read via http(s) and that we will proceed to show the ends up in a a lot of decipherable HTML tabular fashion.

II.4. SAPUI5

SAP UI development toolkit for HTML5 (SAPUI5) could be a client-side cross-browser JavaScript library for building fashionable wealthy net applications. it's JavaScript OpenAjax-compliant and it absolutely supports SAP's product standards. it's protractile, light-weight and straightforward to consume and may be combined with 3rd-party JS libraries. As is customary with SAPUI5 applications the UI5 JavaScript libraries have to be compelled to be loaded before they'll be used, victimization the bootstrap notation inform to the right location among the script tag.

```
1  <html>
2  <head>
3    <title>COPA: Planned vs Actual Sales</title>
4    <script id="sap-ui-bootstrap"
5        type="text/javascript"
6        src="/sap/ui5/1/resources/sap-ui-core.js"
7        data-sap-ui-theme="sap_goldreflection"
8        data-sap-ui-libs="sap.ui.ux3,sap.ui.commons,
9        sap.service.visualization, sap.ui.table">
10     </script>
11
```

Fig. 8. SAPUI5 Library

The data-sap-ui-libs declaration is employed to import different SAPUI5 functions and objects. The Navigator read may be wont to verify the installation and placement of SAPUI5.

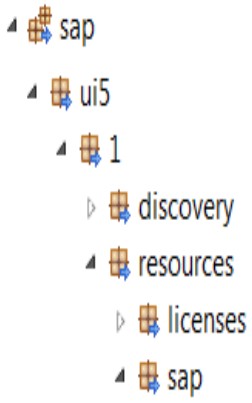


Fig. 9. SAPUI5 Library in Navigator read package structure

Hint: a decent supply of data concerning SAPUI5 may be found domestically on the SAP HANA server.

Fig. 10. SAPUI5 API

The next step needs attention to detail- it's necessary to know the handshaking choices out there between the client-side JavaScript code and also the server-side service. Since a lot of of the heavy-lifting cryptography has already been done behind the scenes among SAPUI5, you'll merely faucet into the framework which can considerably speed up your development effort. There square measure many choices out there looking on the info format and protocol used.

```
12 <script>
13 function popColumns(names) {
14   for (var i=0; i<names.length; i++) {
15     var oColumn = new Column({
16       label: new Label({text: names[i]}),
17       template: new TextView().bindProperty("text",
18       names[i]), width: "100px" });
19     oTable.addColumn(oColumn);
20     }
21 }
22 var url = "../odata/PlannedvsActualCalculationView.xsodata/";
23 var odataModel = new sap.ui.model.odata.ODataModel(url, false);
24 sap.ui.getCore().setModel(odataModel);
25
26 var oTable = new sap.ui.table.Table(
27     { title: "COPA", visibleRowCount: 10, firstVisibleRow: 3});
28 oTable.setColumnHeaderHeight(20);
```

Fig. 11.Client-side Java Script victimization SAPUI5

Line 23: the primary step in rendering Associate in nursing information victimization SAPUI5 is to form an instance of a selected model. SAPUI5 implements strict MVC patterns whereby the modelretrieves and permits for data binding to UI parts. many predefined models area unit offered in SAPUI5; this paper has examples for the OData, JSON models.

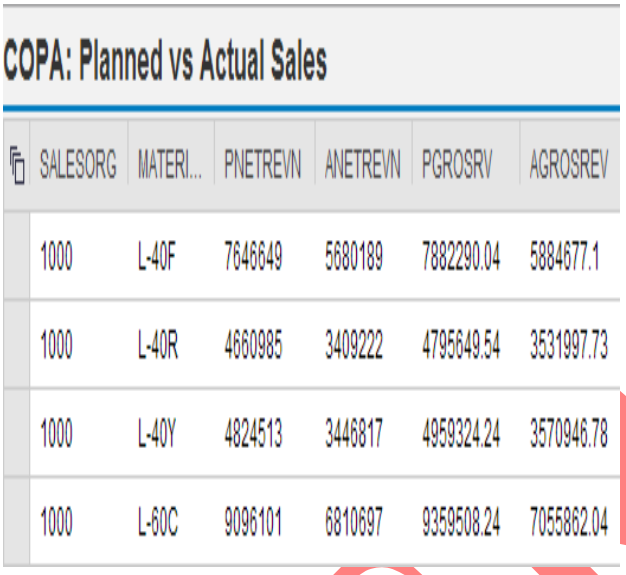
The OData model may be a server-side model, thus solely the information that's requested by the UI is loaded from the server; any amendment of binding or list operations needs a replacement request to the server. The OData model's builder consists of 1 needed parameter that is that the URI inform to the foundation of the OData Service Definition Language file location.

Line 24: Exposes the OData Model to the SAPUI5 libraries and makes it well-known to the framework.

Line 30: This line isn't important; it's a fast and straightforward thanks to minimize the secret writing by employing a custom helper operate to loop through the desired columns.

Line 34 and 39. The OData model is absolute to the table victimization the gathering property found with the OData Service Definition Language file. Notice the choose parameter used earlier.

After successful activation and readying the CO-PA Planned vs Actual Sales report is displayed showing the product kind of like what was displayed earlier victimization the SAP HANA Studio SQL Editor. However, the instance isn't complete; the product got to be filtered.



COPA: Planned vs Actual Sales

SALESORG	MATERI...	PNETREVN	ANETREVN	PGROSRV	AGROSREV
1000	L-40F	7646649	5680189	7882290.04	5884677.1
1000	L-40R	4660985	3409222	4795649.54	3531997.73
1000	L-40Y	4824513	3446817	4959324.24	3570946.78
1000	L-60C	9096101	6810697	9359508.24	7055862.04

Fig. 12. Information result rendered victimization SAPUI5

II.5. JSON

Typical, business necessities demand bigger flexibility via runtime parameter choice typically to limit the information. it's invariably easier to 1st establish the actual uniform resource locator question parameter before embarking on the side secret writing. This additionally provides a glimpse of however the JSON information is structured by appending the format=json parameter.

Hint: Filtering is achieved by introducing the <$filter> and parameters as follows.

>>copaxs/odata/PlannedvsActualCalculationView.xsodata/PlannedvsActualCalcView?$select=MATERIAL,ANETREVN&$filter=startswith(MATERIAL,"DPC')&$inlinecount=allpages&$format=json

Hint: The parameter may be wont to verify the proper variety of expected results.

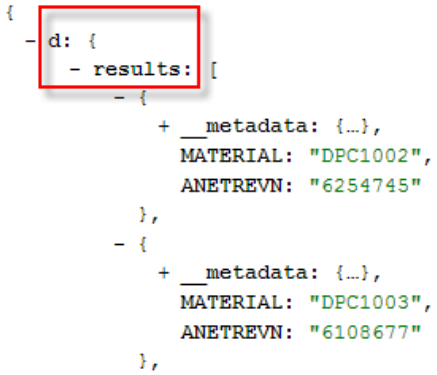


Fig. 13.OData with filtering in JSON format

Hint: observe of the document root component together with the kid component. Those parts are going to be documented later once binding the information victimization SAPUI5.In distinction to the OData model the JSON model may be a client-side model, the information of the model is loaded utterly and is offered on the consumer. it's thus meant for tiny datasets, that area unit utterly offered on the consumer, it doesn't contain any mechanism for server primarily based paging. Notice the <$select> clause that contains all the desired column names together with the parameter.

```
<script>
var jasonModel = new sap.ui.model.json.JSONModel();
var url = "../odata/PlannedvsActualCalculationView.xsodata/
  PlannedvsActualCalcView/?$select=SALESORG,MATERIAL,PNETREVN,
  ANETREVN,PGROSRV,AGROSREV,PSALESDC,ASALESDC,PLANDCM1,
  ACTUACM1,PLANDCM2,ACTUACM2
  &$filter=startswith(MATERIAL,'DPC')";
jasonModel.loadData(url,{},false, "GET");
sap.ui.getCore().setModel(jasonModel);
oTable.setModel(jasonModel);
```

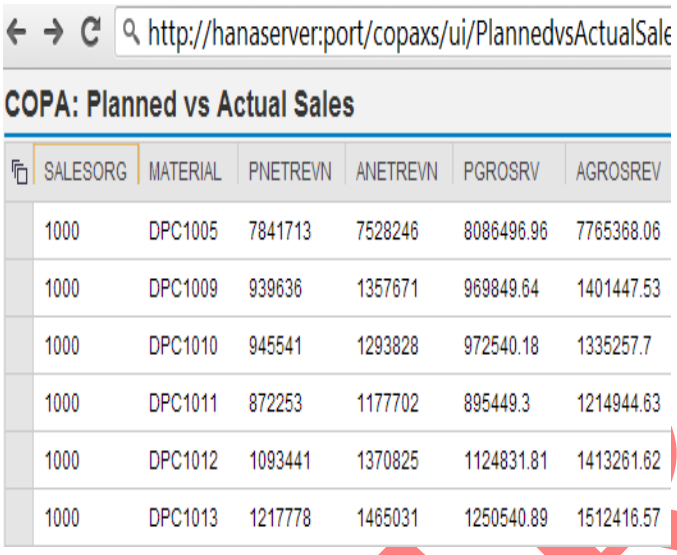
Fig. 14. JavaScript victimization JSONModel in SAPUI5

```
    oTable.setModel(jasonModel);
    oTable.bindRows("/d/results", oTable);
    oTable.placeAt(content);
```

Fig. 15.Binding results victimization JSON Model

Line 17: Group and combines the JSON model to the table UI element.

Line 18: Combines the result set rows to the table victimization the parts seen earlier among the JSON document.



SALESORG	MATERIAL	PNETREVN	ANETREVN	PGROSRV	AGROSREV
1000	DPC1005	7841713	7528246	8086496.96	7765368.06
1000	DPC1009	939636	1357671	969849.64	1401447.53
1000	DPC1010	945541	1293828	972540.18	1335257.7
1000	DPC1011	872253	1177702	895449.3	1214944.63
1000	DPC1012	1093441	1370825	1124831.81	1413261.62
1000	DPC1013	1217778	1465031	1250540.89	1512416.57

Fig. 16. Final results aggregate and filtered

The data is aggregate and filtered consistent with the necessities.

CONCLUSION

This paper provided the knowledge to make a sublime next generation period application for ERP that is additional user friendly and responsive style that may work on any platform and devices the utilization of REST primarily based OData technology for net services helps to modify the design, adopt SOA and provides a platform to scale – up net services effectively.

REFERENCES

[1] HassoPlattner, HassoPlattner Institute for IT Systems Engineering, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany, SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA. Copyright 2009 ACM 978-1-60558-551-2/09/06

[2] Blending Transactions And Analytics in single In – Memory Platform: Key to real – time Enterpise, Carl W. Olofson Henry D. Morris, February 2013

[3] Introducing OData, Data Access for the Web, the cloud, mobile devices, and more, David Chappell (Chappell & Associates), Published: May 2011

[4] SAP HANA Developer Guide, SAP HANA Platform SPS 08, Document Version: 1.1 - 2014-08-21, SAP SE or an SAP affiliate company.

[5]SAP User Experience Community, http://experience.sap.com/ux-strategy/, http://www.sapdesignguild.org/