

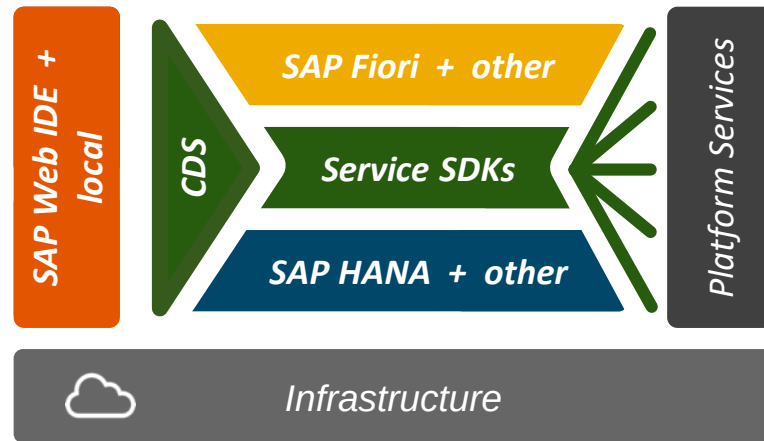


Week 2: New Programming Model

Unit 3: SQL DDL Artifacts

SQL DDL Artifacts

CDS usage within the new application programming model



Task	Using	And let the framework take care of...
Define Data Models	CDS	- Mapping to physical schemas (e.g. SQL) Deployment to SAP HANA + others - Foreign keys and joins - Localized data - Extensibility - Language-embedded data access
Define Services	CDS	- Model translation (e.g. to OData, Open API) - Serving requests out-of-the-box - Standard validations - Auto-filling audit data - Trace and audit logging - Authorization enforcement
Add Custom Logic	Service SDKs for Java or Node.js	- Multitenancy and isolation - Connection pooling - Validating input - Message handling - Serializing responses - Platform integration
Consume Services (e.g. from S/4)		- Providing APIs in Java / Node.js - Mashups with local data - Handling http requests - Resilience, caching, ...
Add UIs	CDS Annotations	- Translating to OData vocabularies - Generating SAP Fiori app skeletons

SQL DDL Artifacts

CDS: database-agnostic syntax

- **Deployment to SAP HANA + others = database-agnostic syntax**
- Although the syntax is similar to the older hdbdd/hdbcds, SAP HANA-specific data types and features are left out
- For example: no partition settings, no identity columns, no direct calculation view consumption, no spatial data types, no SAP HANA-specific SQL functions (some limitations may be lifted over time)

```
18 entity Headers {
19   key PURCHASEORDERID : Integer @(title: '{i18n>po_id}', Common.FieldControl: #Mandatory, Search.defaultSearchElement, Common.Label: '{i18n>po_id}');
20   ITEMS                : association to many Items on ITEMS.POHeader = $self @(title: '{i18n>po_items}', Common: { Text: {$value: ITEMS.PRODUCT, "@UI.TextArrangement": #TextOnly }});
21   HISTORY              : HistoryT;
22   NOTEID               : BusinessKey null @(title: '{i18n>notes}');
23   PARTNER              : BusinessKey @(title: '{i18n>partner_id}');
24   CURRENCY              : CurrencyT @(
25     Common: {
26       Text: {$value: CURRENCY.CURRENCY, "@UI.TextArrangement": #TextOnly},
27       ValueList: {entity: 'CURRENCY', type: #fixed},
28       ValueListWithFixedValues
29     }
30   );
31   GROSSAMOUNT          : AmountT @( title: '{i18n>grossAmount}', Measures.ISOCurrency: currency);
32   NETAMOUNT            : AmountT @( title: '{i18n>netAmount}', Measures.ISOCurrency: currency);
33   TAXAMOUNT            : AmountT @( title: '{i18n>taxAmount}', Measures.ISOCurrency: currency);
34   LIFECYCLESTATUS      : StatusT @(title: '{i18n>lifecycle}', Common.FieldControl: #ReadOnly);
35   APPROVALSTATUS       : StatusT @(title: '{i18n>approval}', Common.FieldControl: #ReadOnly);
36   CONFIRMSTATUS        : StatusT @(title: '{i18n>confirmation}', Common.FieldControl: #ReadOnly);
37   ORDERINGSTATUS       : StatusT @(title: '{i18n>ordering}', Common.FieldControl: #ReadOnly);
38   INVOICINGSTATUS      : StatusT @(title: '{i18n>invoicing}', Common.FieldControl: #ReadOnly);
39 }
40
41
42 entity Items {
43   key POHeader          : association to Headers @title: '{i18n>poService}';
44   key PRODUCT           : BusinessKey @(title: '{i18n>product}', Common.FieldControl: #Mandatory, Search.defaultSearchElement);
45   NOTEID               : BusinessKey null ;
46   CURRENCY              : CurrencyT @(
47     Common: {
48       Text: {$value: CURRENCY.CURRENCY, "@UI.TextArrangement": #TextOnly},
49       ValueList: {entity: 'CURRENCY', type: #fixed},
50       ValueListWithFixedValues
51     }
52   );
53   GROSSAMOUNT          : AmountT @( title: '{i18n>grossAmount}', Measures.ISOCurrency: currency);
54   NETAMOUNT            : AmountT @( title: '{i18n>netAmount}', Measures.ISOCurrency: currency);
55   TAXAMOUNT            : AmountT @( title: '{i18n>taxAmount}', Measures.ISOCurrency: currency);
56   QUANTITY              : QuantityT;
57   QUANTITYUNIT          : UnitT;
58   DELIVERYDATE          : SDate @(title: '{i18n>deliveryDate}');
59 }
```

SQL DDL Artifacts

DDL-based artifacts as alternative to CDS for SAP HANA-specific feature access

- Core Data Services is good for defining reusable types, multiple entities (tables), views, and relationships via associations all within one source file.
- But now that CDS is database-agnostic, it is preferred to use DDL-based artifacts to access SAP HANA-specific features. Since SPS11, with HDI we also have the DDL-based artifacts.
 - Tables
 - Indexes
 - Full-Text Indexes
 - Constraints
 - Triggers
 - Procedures
 - Functions
 - SQL Views
 - Table Types
 - Sequences
 - Statistics
 - Structured Privilege
 - Graph Workspace
 - Collections
 - Result Cache

SQL DDL Artifacts

Tables

- The table plugin .hdbtable transforms a design-time table resource into a table database object

```
Header.hdbtable x
1 COLUMN TABLE "PO.Header" (
2     "PURCHASEORDERID" INTEGER CS_INT GENERATED BY DEFAULT AS IDENTITY (NO CYCLE NO CACHE NO MINVALUE START WITH 300001000 INCREMENT BY 1 MAXVALUE 399999999) NOT NULL COMMENT 'Purchase Order ID',
3     "HISTORY.CREATEDBY.EMPLOYEEID" INTEGER CS_INT COMMENT 'Created By',
4     "HISTORY.CREATEDAT" DATE CS_DAYDATE COMMENT 'Created Date',
5     "HISTORY.CHANGEDBY.EMPLOYEEID" INTEGER CS_INT COMMENT 'Last Updated By',
6     "HISTORY.CHANGEDAT" DATE CS_DAYDATE COMMENT 'Last Update Date',
7     "NOTEID" NVARCHAR(10) COMMENT 'Notes',
8     "PARTNER.PARTNERID" INTEGER CS_INT COMMENT 'Partner Association',
9     "CURRENCY" NVARCHAR(5) COMMENT 'Currency',
10    "GROSSAMOUNT" DECIMAL(15,2) CS_FIXED COMMENT 'Gross Amount',
11    "NETAMOUNT" DECIMAL(15,2) CS_FIXED COMMENT 'Net Amount',
12    "TAXAMOUNT" DECIMAL(15,2) CS_FIXED COMMENT 'Tax Amount',
13    "LIFECYCLESTATUS" NVARCHAR(1) COMMENT 'Lifecycle Status',
14    "APPROVALSTATUS" NVARCHAR(1) COMMENT 'Approval Status',
15    "CONFIRMSTATUS" NVARCHAR(1) COMMENT 'Confirmation Status',
16    "ORDERINGSTATUS" NVARCHAR(1) COMMENT 'Ordering Status',
17    "INVOICINGSTATUS" NVARCHAR(1) COMMENT 'Invoicing Status',
18    PRIMARY KEY ("PURCHASEORDERID"))
19    WITH ASSOCIATIONS( JOIN "MD.Employees" AS "HISTORY.CHANGEDBY" ON "HISTORY.CHANGEDBY"."EMPLOYEEID" = "HISTORY.CHANGEDBY.EMPLOYEEID",
20                      JOIN "MD.Employees" AS "HISTORY.CREATEDBY" ON "HISTORY.CREATEDBY"."EMPLOYEEID" = "HISTORY.CREATEDBY.EMPLOYEEID",
21                      JOIN "PO.Item" AS "ITEMS" ON "PURCHASEORDERID" = "PURCHASEORDERID",
22                      JOIN "MD.BusinessPartner" AS "PARTNER" ON "PARTNER"."PARTNERID" = "PARTNER.PARTNERID")
23    COMMENT 'Purchase Order Header'
24    UNLOAD PRIORITY 5 AUTO MERGE
```

SQL DDL Artifacts

Indexes

- The index plugin .hdbindex transforms a design-time index resource into an index on a database table

partnerPO.hdbindex x	
1	INDEX "PARTNER_IDX" ON "PurchaseOrder.Header" (PARTNER)

SQL DDL Artifacts

Full-text index

- The full-text index plugin .hdbfulltextindex transforms a design-time full-text index resource into a full-text index on a database table

CUSTOMER_NAME_FTI.hdbf... x	
1	FULLTEXT INDEX "CUSTOMER_NAME_FTI" ON "Customers" (NAME)

SQL DDL Artifacts

Constraints

- The foreign-key constraint plugin .hdbconstraint transforms a design-time constraint into a constraint on database tables

PO_CONST.hdbconstraint x

```
1 CONSTRAINT PO_CONSTRAINT
2 ON "PO.Header" FOREIGN KEY ("PARTNER.PARTNERID")
3 REFERENCES "MD.BusinessPartner" ("PARTNERID") ON UPDATE CASCADE
```

SQL DDL Artifacts

Trigger

- The trigger plugin .hdbtrigger transforms a design-time trigger resource into a trigger on a database table

```
add_insert_to_log.hdbtrigger x
1  TRIGGER "add_insert_to_log"
2      AFTER INSERT ON
3      "Product.Products"
4      REFERENCING NEW ROW newrow FOR EACH ROW
5  BEGIN
6
7      INSERT INTO "Product.ProductLog"
8          VALUES(:newrow."PRODUCTID",
9              (select IFNULL(MAX(logid), 0) + 1 from
10                  "Product.ProductLog" where productid = :newrow.PRODUCTID),
11              now(),
12              CURRENT_USER,
13              :newrow."PRODUCTID" || ' has been created');
14  END;
```

SQL DDL Artifacts

SQL views

- The view plugin .hdbview transforms a design-time view resource into a SQL view database object

```
Header.hdbview x
1  VIEW "PO.HeaderView" COMMENT 'Purchase Order Header View'
2    ( "PurchaseOrderId",
3      "CreatedByEmployeeId",
4      "CreatedByFirstName",
5      "CreatedByLastName",
6      "CreatedByLoginName",
7      "CreatedAt",
8      "PartnerId",
9      "CompanyName",
10     "Currency",
11     "GrossAmount",
12     "NetAmount",
13     "TaxAmount" ) AS SELECT
14     "Header_$0"."PURCHASEORDERID" AS "PurchaseOrderId" ,
15     "Header_$0"."HISTORY.CREATEDBY"."EMPLOYEEID" AS "CreatedByEmployeeId" ,
16     "Header_$0"."HISTORY.CREATEDBY"."NAME.FIRST" AS "CreatedByFirstName" ,
17     "Header_$0"."HISTORY.CREATEDBY"."NAME.LAST" AS "CreatedByLastName" ,
18     "Header_$0"."HISTORY.CREATEDBY"."LOGINNAME" AS "CreatedByLoginName" ,
19     "Header_$0"."HISTORY.CREATEDAT" AS "CreatedAt" ,
20     "Header_$0"."PARTNER"."PARTNERID" AS "PartnerId" ,
21     "Header_$0"."PARTNER"."COMPANYNAME" AS "CompanyName" ,
22     "Header_$0"."CURRENCY" AS "Currency" ,
23     "Header_$0"."GROSSAMOUNT" AS "GrossAmount" ,
24     "Header_$0"."NETAMOUNT" AS "NetAmount" ,
25     "Header_$0"."TAXAMOUNT" AS "TaxAmount"
26 FROM "PO.Header" AS "Header_$0" WITH READ ONLY
```

SQL DDL Artifacts

Table types

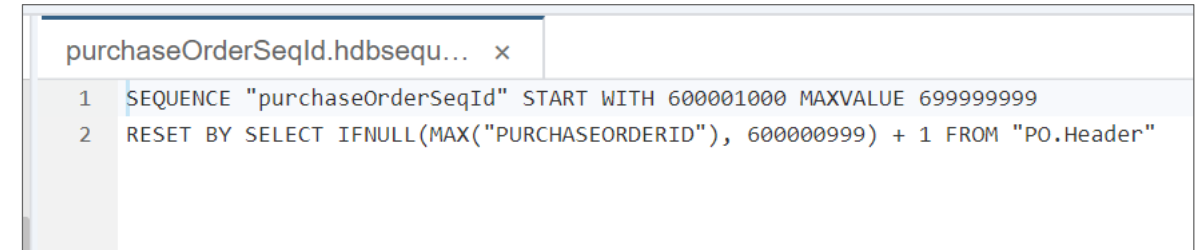
- The table type plugin .hdbtabletype transforms a design-time table type resource into a table type database object, for example, for use by signatures of SQL procedures

```
tt_bp_addresses.hdbtabletype x
1 TYPE "Procedures.tt_bp_addresses" AS TABLE (
2     ....
3     "PARTNERID" NVARCHAR(10) CS_STRING,
4     "PARTNERROLE" NVARCHAR(3) CS_STRING,
5     "EMAILADDRESS" NVARCHAR(255) CS_STRING,
6     "COMPANYNAME" NVARCHAR(80) CS_STRING,
7     "ADDRESSID" NVARCHAR(10) CS_STRING,
8     "CITY" NVARCHAR(40) CS_STRING,
9     "POSTALCODE" NVARCHAR(10) CS_STRING,
10    "STREET" NVARCHAR(60) CS_STRING )
```

SQL DDL Artifacts

Sequences

- The sequence plugin .hdbsequence transforms a design-time sequence resource into a sequence database object. If the sequence definition contains a RESET BY query, then this RESET BY query is executed during deployment to set the sequence to its start value. Sequences without a RESET BY query are also supported



The screenshot shows a code editor window with a tab titled "purchaseOrderSeqId.hdbsequ...". The editor contains two lines of SQL code:

```
1 SEQUENCE "purchaseOrderSeqId" START WITH 600001000 MAXVALUE 699999999
2 RESET BY SELECT IFNULL(MAX("PURCHASEORDERID"), 600000999) + 1 FROM "PO.Header"
```

SQL DDL Artifacts

Statistics

- The statistics plugin .hdbstatistics transforms a design-time statistics resource into a statistics object on a database table

```
STATISTICS "com.sap.hana.example::CUSTOMER_NAME_STATISTICS"  
ON "com.sap.hana.example::CUSTOMERS" (NAME)
```

SQL DDL Artifacts

Structured privilege

- The structured privilege plugin .hdbstructuredprivilege transforms a design-time DDL-based structured privilege resource into a structured privilege object in the database. The referenced views must be defined using the WITH STRUCTURED PRIVILEGE CHECK clause

PurchaseOrder.hdbstructuredp... x	
1	STRUCTURED PRIVILEGE
2	"PO_VIEW_PRIVILEGE"
3	FOR SELECT ON
4	"PurchaseOrder.ItemView"
5	WHERE "CurrencyCode" = 'EUR'

SQL DDL Artifacts

Graph workspace

- The graph workspace plugin .hdbgraphworkspace transforms a design-time graph workspace resource into a graph workspace object in the database

```
GRAPH WORKSPACE MY_GRAPH_WORKSPACE
EDGE TABLE THE_EDGE_TABLE
    SOURCE COLUMN SOURCE TARGET COLUMN TARGET KEY COLUMN EDGE_ID
VERTEX TABLE THE_VERTEX_TABLE
    KEY COLUMN VERTEX_ID
```

SQL DDL Artifacts

Collections

- The collection plugin .hdbcollections transforms a design-time, document-collection resource into a collection database object, which is a data store with the table type "COLLECTION". The so-called "collections" are used to store JSON documents in the SAP HANA document store (DocStore)

COLLECTION TABLE "CUSTOMERS"

SQL DDL Artifacts

Result cache

- The result cache plugin .hdbresultcache transforms a design-time DDL-based definition of a result cache into a catalog-level result cache definition. The file must use an artificial cache name which uses the prefix `_SYS_CACHE#` followed by the name of the referenced view or function

```
RESULT CACHE "_SYS_CACHE#com.sap.hana.example::A_SQL_View"  
ON VIEW "com.sap.hana.example::A_SQL_View"  
WITH RETENTION 30
```

SQL DDL Artifacts

CDS: wrappers for existing DB artifacts

- From CDS you can create a “wrapper/phantom” entity for an existing DB artifact using the **@cds.persistence.exists** annotation
- You must completely redefine the entire entity in CDS with matching names (include case)
- In the future, we plan to add a kind of import option to avoid this painful redefinition
- This approach must be used to service-enable calculation views
- Synonyms can be used to shorten the names and make them CDS-compatible

```
@cds.persistence.exists
Entity BUYER {
  BUILDING: String(10) @title: '{i18n>building}';
  CITY: String(40) @title: '{i18n>city}';
  COMPANYNAME: String(80) @title: '{i18n>company}';
  COUNTRY: String(3) @title: '{i18n>country}';
  EMAILADDRESS: String(255) @title: '{i18n>email}';
  LEGALFORM: String(10) @title: '{i18n>legal}';
  key PARTNERID: Integer @title: '{i18n>partnerId}';
  PARTNERROLE: String(3) @title: '{i18n>partnerRole}';
  POSTALCODE: String(10) @title: '{i18n>postalCode}';
  REGION: String(4) @title: '{i18n>region}';
  STREET: String(60) @title: '{i18n>street}';
}
```

local.hdbsynonym x	
+	
Synonym Name	Object Name
☐ BUYER	core.models::BUYER ...
☐ <Click to Add>	

BUYER.hdbcalculationview x

Semantics

Projection

Join_1

MD.Addresses

MD.BusinessPartner

100%

Semantics

View Properties Columns (11) Hierarchies (0) Parameters (0)

Private

Type	Name	Label	Aggregation...
☐	PARTNERID	Partner ID	
☐	PARTNERROLE	Partner Role	
☐	EMAILADDRESS	Email Address	
☐	COMPANYNAME	Company Name	
☐	LEGALFORM	Legal Form	
☐	CITY	City	
☐	POSTALCODE	Postal Code	
☐	BUILDING	Building	
☐	STREET	Street	
☐	COUNTRY	Country	
☐	REGION	Region	

Thank you.

Contact information:

open@sap.com

Follow all of SAP



www.sap.com/contactsap

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See www.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.