Week 3 Unit 1: **Recent Open SQL Enhancements – Part 1**

Recent Open SQL Enhancements – Part 1
Outline
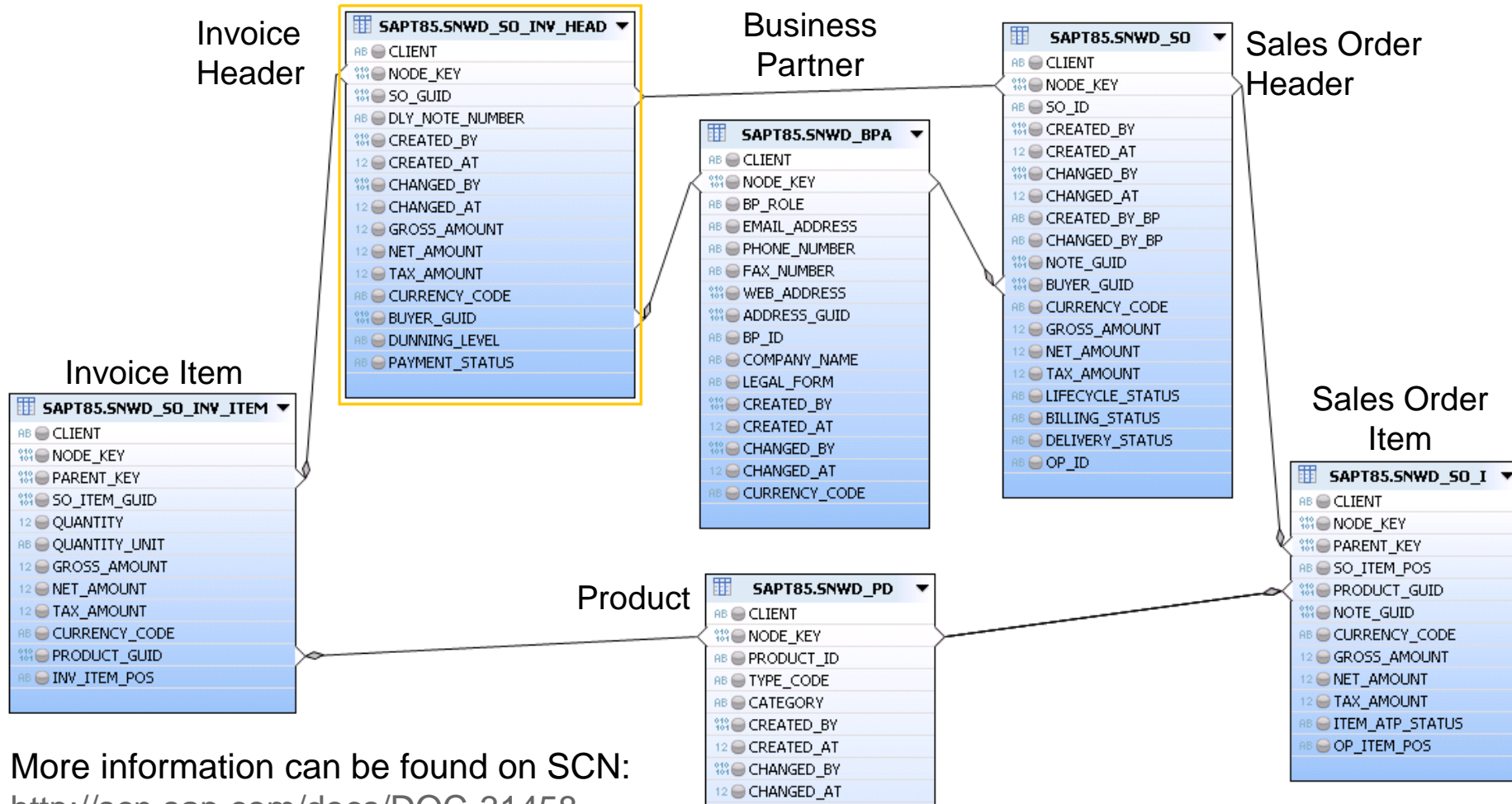
Content

- Enterprise Procurement Model

- Reminder: What is Open SQL?

- Open SQL evolution

- Open SQL enhancements
  - Syntax
  - SELECT list

Public

Recent Open SQL Enhancements – Part 1
Enterprise Procurement Model (EPM)

Invoice Header

SAPT85.SNWD_SO_INV_HEAD
- CLIENT
- NODE_KEY
- SO_GUID
- DLY_NOTE_NUMBER
- CREATED_BY
- CREATED_AT
- CHANGED_BY
- CHANGED_AT
- GROSS_AMOUNT
- NET_AMOUNT
- TAX_AMOUNT
- CURRENCY_CODE
- BUYER_GUID
- DUNNING_LEVEL
- PAYMENT_STATUS

Business Partner

SAPT85.SNWD_BPA
- CLIENT
- NODE_KEY
- BP_ROLE
- EMAIL_ADDRESS
- PHONE_NUMBER
- FAX_NUMBER
- WEB_ADDRESS
- ADDRESS_GUID
- BP_ID
- COMPANY_NAME
- LEGAL_FORM
- CREATED_BY
- CREATED_AT
- CHANGED_BY
- CHANGED_AT
- CURRENCY_CODE

Sales Order Header

SAPT85.SNWD_SO
- CLIENT
- NODE_KEY
- SO_ID
- CREATED_BY
- CREATED_AT
- CHANGED_BY
- CHANGED_AT
- CREATED_BY_BP
- CHANGED_BY_BP
- NOTE_GUID
- BUYER_GUID
- CURRENCY_CODE
- GROSS_AMOUNT
- NET_AMOUNT
- TAX_AMOUNT
- LIFECYCLE_STATUS
- BILLING_STATUS
- DELIVERY_STATUS
- OP_ID

Invoice Item

SAPT85.SNWD_SO_INV_ITEM
- CLIENT
- NODE_KEY
- PARENT_KEY
- SO_ITEM_GUID
- QUANTITY
- QUANTITY_UNIT
- GROSS_AMOUNT
- NET_AMOUNT
- TAX_AMOUNT
- CURRENCY_CODE
- PRODUCT_GUID
- INV_ITEM_POS

Product

SAPT85.SNWD_PD
- CLIENT
- NODE_KEY
- PRODUCT_ID
- TYPE_CODE
- CATEGORY
- CREATED_BY
- CREATED_AT
- CHANGED_BY
- CHANGED_AT

Sales Order Item

SAPT85.SNWD_SO_I
- CLIENT
- NODE_KEY
- PARENT_KEY
- SO_ITEM_POS
- PRODUCT_GUID
- NOTE_GUID
- CURRENCY_CODE
- GROSS_AMOUNT
- NET_AMOUNT
- TAX_AMOUNT
- ITEM_ATP_STATUS
- OP_ITEM_POS

More information can be found on SCN:
http://scn.sap.com/docs/DOC-31458

Recent Open SQL Enhancements – Part 1
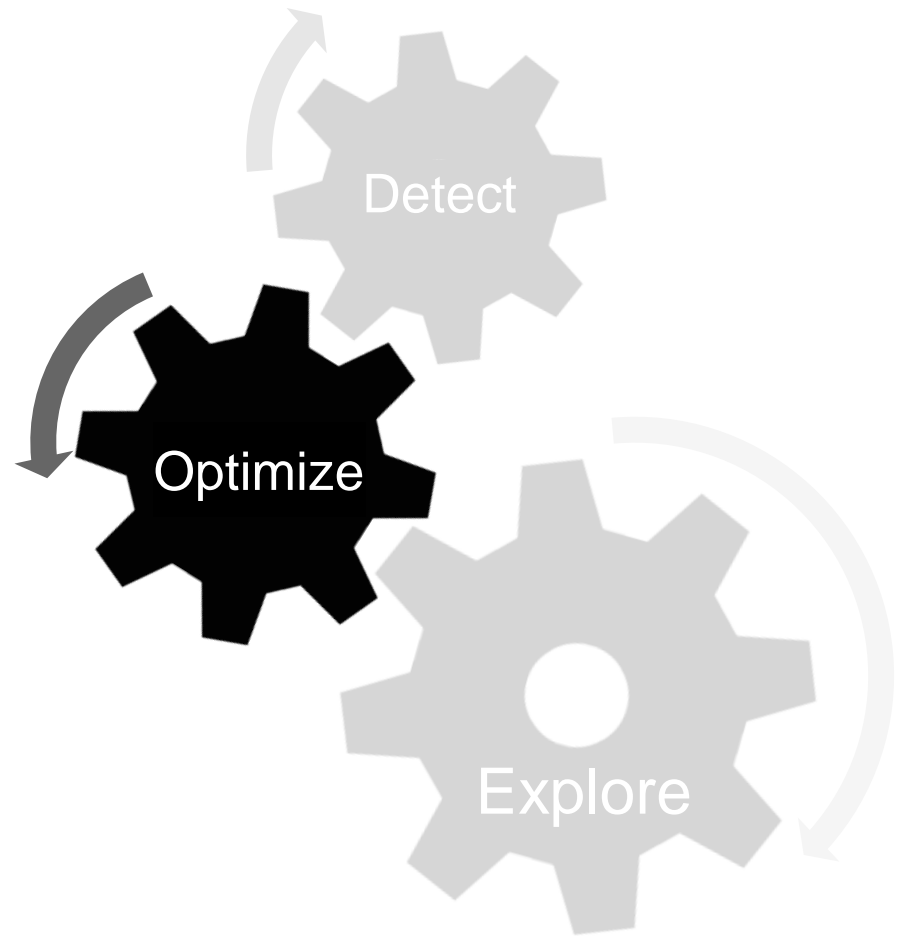## Migration to SAP HANA

1. Detect

- Functional correctness
- Performance optimization potential

2. Optimize

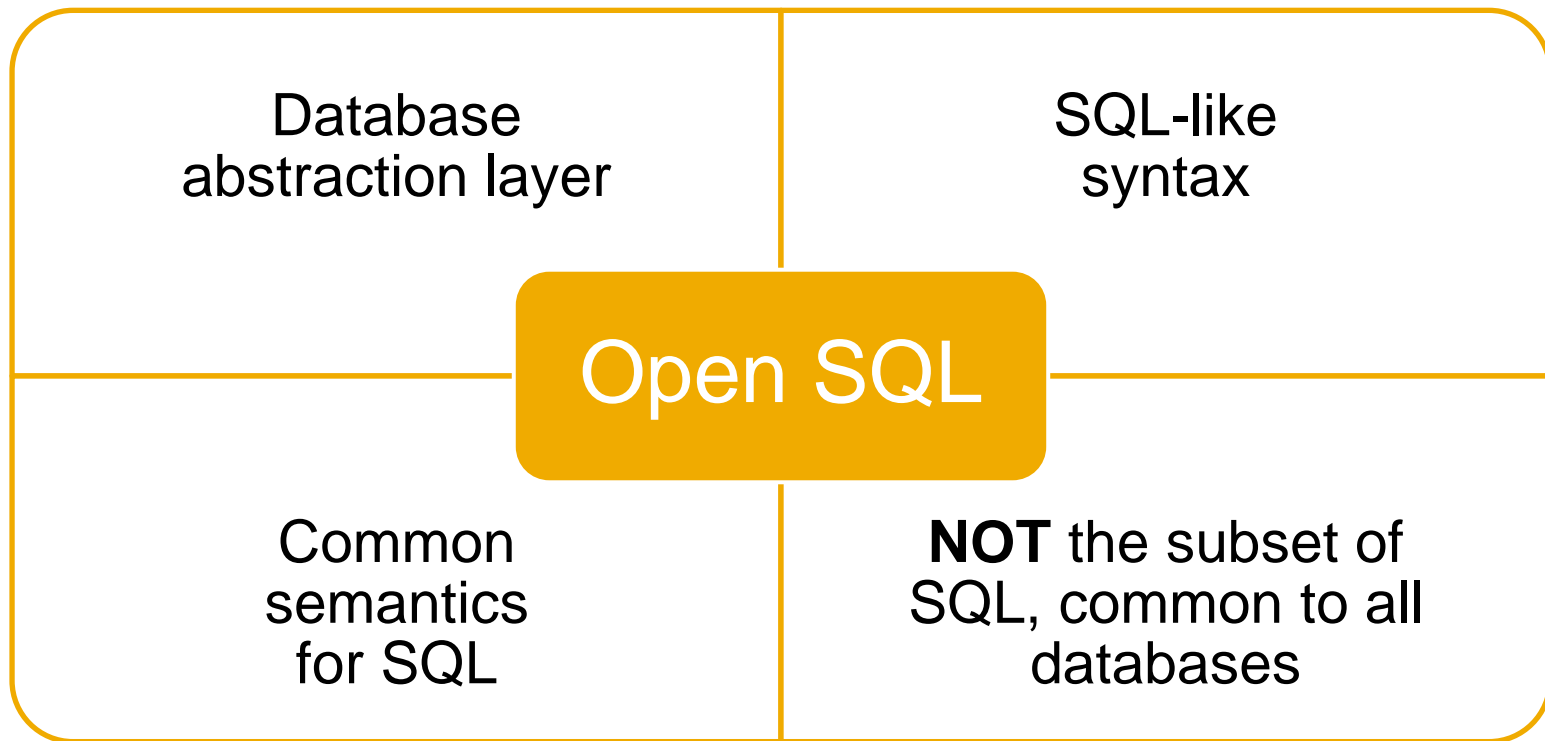- Database-oriented programming

3. Explore

- Use SAP HANA-specific features
- Rethink & innovate

Detect

Optimize

Explore

Recent Open SQL Enhancements – Part 1

Reminder: What Is Open SQL?

Database abstraction layer	SQL-like syntax
	Open SQL	
Common semantics for SQL	**NOT** the subset of SQL, common to all databases

Open SQL is the only DB **abstraction layer** that defines a **common semantic** for all SAP-supported databases!

Recent Open SQL Enhancements – Part 1
Open SQL in ABAP 7.4 SP2 and Beyond

Open SQL aspires to

- enable the application of the Code-to-Data paradigm

- provide (more) standard SQL features

- enable the consumption of SAP HANA-specific features

Introduction of new features

Reduction of existing limitations

SQL92 Standard

Open SQL
in ABAP 7.4 SP2 and beyond

Open SQL
(releases before ABAP 7.4)
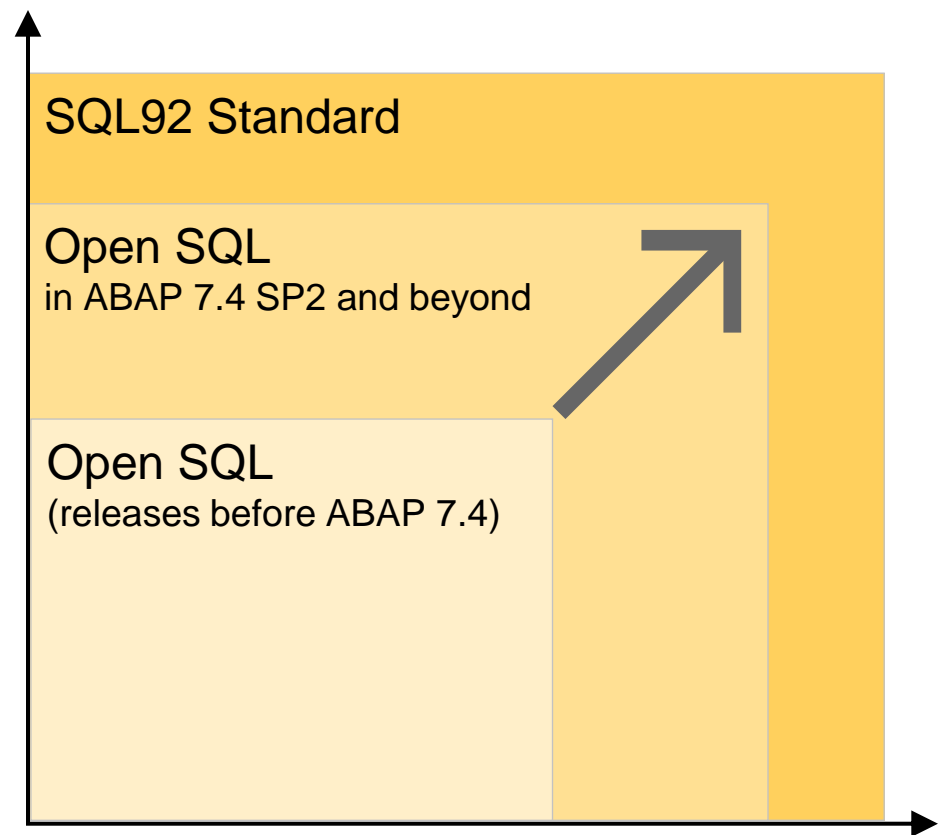
Recent Open SQL Enhancements – Part 1
Open SQL: Demo

Open SQL enhancements

- Syntax enhancements:
  - Escaping of host variables
  - Comma-separated select list
- SELECT list enhancements:
  - Aggregation functions
  - Literals
  - Arithmetical expressions

Public   7
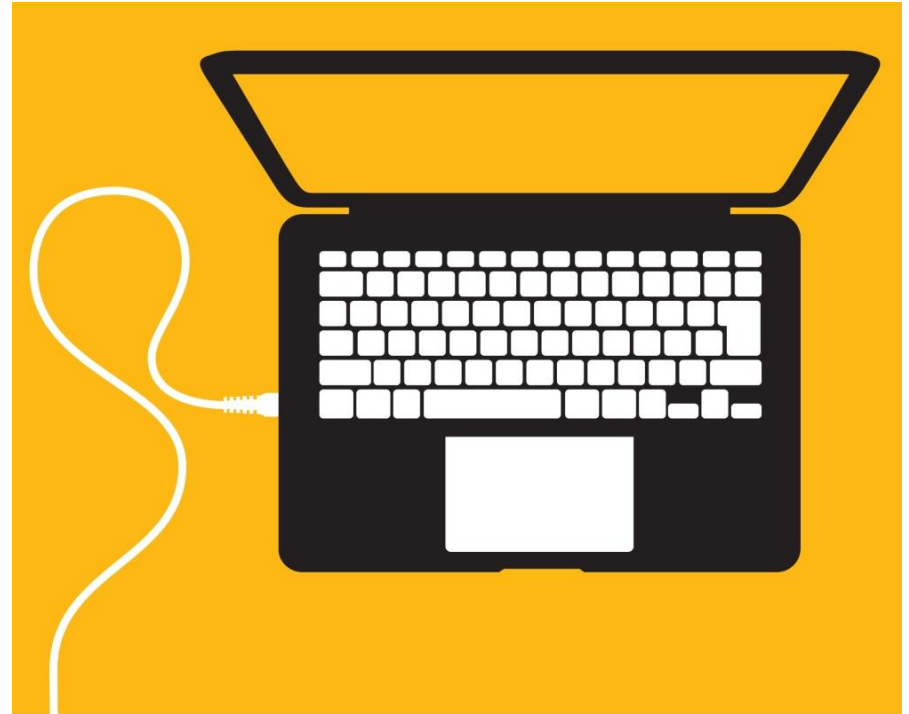
Recent Open SQL Enhancements – Part 1
What's New in Open SQL?

New Open SQL Syntax

- Escaping of host variables
- Comma separated element list

Target Type Inference

New SELECT List Features

- Aggregation functions
- Literal values (next slide)
- Arithmetic expressions (next slide)
- String expression (next slide)

```
SELECT so_id,
       currency_code,
       gross_amount
  FROM snwd_so
  INTO TABLE @DATA(lt_result).
```

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
           AS total_gross_amount
  FROM snwd_so AS so
  INNER JOIN snwd_bpa AS bpa
         ON so~buyer_guid = bpa~node_key
  INTO TABLE @DATA(lt_result)
  GROUP BY bp_id, company_name,
           so~currency_code.
```

Literal Values

- Can now be used in the SELECT list
- Allow for a generic implementation of an existence check

```abap
SELECT so~so_id,
       'X' AS literal_x,
       42 AS literal_42
   FROM snwd_so AS so
   INTO TABLE @DATA(lt_result).
```

```abap
DATA lv_exists TYPE abap_bool
                   VALUE abap_false.

SELECT SINGLE @abap_true
   FROM snwd_so
   INTO @lv_exists.

IF lv_exists = abap_true.
   "do some awesome application logic
ELSE.
   "no sales order exists
ENDIF.
```

Expressions

Arithmetic Expressions

- **+, -, *, DIV, MOD, ABS, FLOOR, CEIL**

- **Remember:** Open SQL defines a semantic for these expressions common to all supported databases

- Refer to the ABAP documentation to see which expression is valid for which types

String Expressions

- Concatenate character columns with the && operator

```abap
DATA lv_discount TYPE p LENGTH 1 DECIMALS 1
                   VALUE '0.8'.

SELECT ( 1 + 1 ) AS two,
       ( @lv_discount * gross_amount )
           AS red_gross_amount,
       CEIL( gross_amount )
           AS ceiled_gross_amount
  FROM snwd_so
  INTO TABLE @DATA(lt_result).
```

```abap
SELECT company_name
    && ' (' && legal_form && ')'
  FROM snwd_bpa
  INTO TABLE @DATA(lt_result).
```

Recent Open SQL Enhancements – Part 1
What's Next?

Week 3 Unit 2
Recent Open SQL Enhancements – Part 2

Public

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG or an SAP affiliate company. All rights reserved.