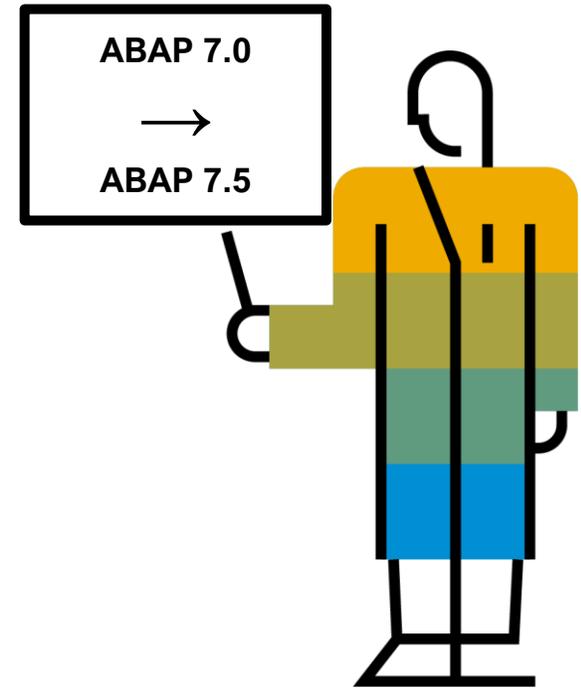


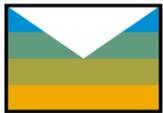
Обучение АВАР. Дельта от 7.0 к 7.5

Василий Ковальский, SAP CIS
2 июня 2021 12:00 – 13:00

PUBLIC



Василий Ковальский,
инструктор АВАР с 1998 года



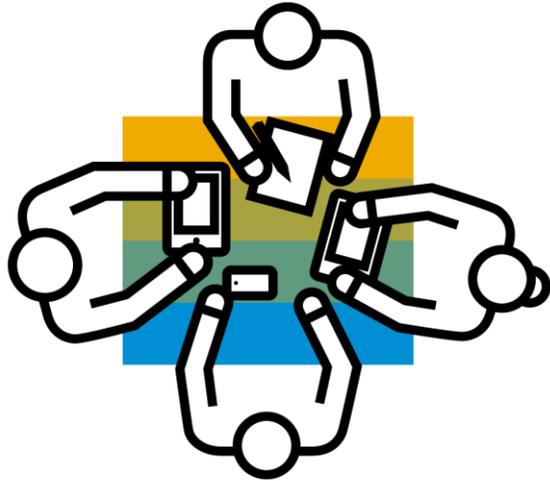
vassili.kovalski@sap.com



<https://www.facebook.com/Vassili.Kovalski>



Для кого этот вебинар?



- для ABAP разработчиков
- для специалистов службы поддержки
- для руководителей подразделений разработки
- для консультантов, сталкивающихся с ABAP кодом или общающихся с разработчиками, в особенности для тех, кто пишут спецификации на разработку
- для специалистов по подготовке и найму персонала

О чем пойдет речь 1

- Строковая и символьная обработка
- Числа и арифметика
- Логические стэйтменты, операторы, функции; предикатный вызов методов
- Инлайновые декларации
- Операторы-конструкторы
- Внутренние таблицы: вторичные ключи, индексные и ключевые выражения, функции, копирование, фильтры, редукция, группировка, итоги по группировке, мэши

О чем пойдет речь 2

- ООП. «Возобновляемые исключения»
- Новый синтаксис Open SQL
- АВАР-каналы
- Словарь. Платформенно-специфичные свойства, глобальные временные таблицы, «опережающий» search-help
- Core Data Services: DDL, DCL, расширения.

Строковая и символьная обработка

- Инлайновая конкатенация. Оператор `&&` . Результат всегда типа `string`
- Строковые шаблоны задаются в модульных скобках `||`
- Строковые выражения в строковых шаблонах задаются в фигурных скобках `{ }`

```
txt = ||{ `ABAP` WIDTH = 16 ALIGN = RIGHT PAD = `=` CASE = UPPER }||.
```

Опции строковых выражений

WIDTH – Длина

DECIMALS – Количество десятичных знаков

ALIGN – Выравнивание

PAD – Заполнение пустых мест

CASE – Регистр

SIGN – Знак

EXPONENT – Экспоненциальная форма

DECIMALS – Количество знаков после десятичного разделителя

ZERO – Вывод нулевого значения

STYLE – Стиль вывода числовых данных

NUMBER – Форматирование разделителей в числах

DATE – Форматирование даты

TIME – Форматирование времени

TIMESTAMP – Форматирование метки времени

TIMEZONE – Форматирование времени .по часовому поясу

COUNTRY – Форматирование по локальным правилам страны

ALPHA – Добавление / удаление ведущих нулей

XSD – Приведение к XML-формату данных

Функция различения строк

Функция различия строк. Возвращает целое число.

distance(val1 = txt1 val2 = txt2 max = mx)

Возвращает значение редакционного расстояния (расстояния Левенштейна),

Функции поиска в строках

find(val = txt sub = sub case = case off = off len = len occ = occ)

Определяет смещение от начала строки, начиная с которого подстрока sub входит в текст txt

find_end(val = txt1 sub = subs case = case off = off len = ln occ = occ)

Аналогично find(...), но ведет поиск не от начала к концу, а от конца к началу

find_any_of(val = txt sub = sub off = off len = ln occ = occ)

Определяет смещение от начала строки txt первого попавшегося символов подстроки subs

find_any_not_of(val = txt sub = sub off = off len = len occ = occ)

Определяет смещение от начала строки txt первого попавшегося символов, который не входит в подстроку subs

Функции преобразования строк

cmax(val1 = txt1 val2 = txt2)

cmin(val1 = txt1 val2 = txt2)

condense(val = txt del = del from = from to = to)

escape(val = txt format = format)

insert(val = txt sub = sub off = off)

match(val = txt regex = regex case = case off = off).

repeat(val = text occ = occ)

replace() (несколько вариантов задания параметров)

reverse(val = text)

segment(val = txt index = index sep = sep).

shift_left(val = txt places = places circular = circular sub = sub)

shift_right(val = txt places = places circular = circular sub = sub)

Функции преобразования строк

substring(val = txt off = off len = len)

substring_*(val = txt sub = sub case = case occ = occ len = len)

to_upper(val = text)

to_lower(val = text)

to_mixed(val = text sep = sep case = case min = min)

from_mixed(val = text sep = sep case = case min = min)

translate(val = text from = from to = to)

Числа и арифметика

Типы данных **decfloat16** и **decfloat34** (с версии 7.02) **int8** (с версии 7.50)
Соответствующие новые методы `get_*` и константы в RTTS

Команда **COMPUTE EXACT** (с версии 7.02, устарела с версий 7.4)

Числовые функции.

round(...) (с версии 7.02)

ipow(base = arg exp = power) (с версии 7.40 SP2)

В принципе функция `ipow()` подобна оператору возведения в степень «**», однако есть нюанс: производительность. В малые степени функция `ipow()` обычно возводит несколько быстрее, чем оператор `**`.

Логические команды и операторы

IF, ELSEIF (с версии 7.02) В командах IF и ELSIF допустимы арифметические и символьные выражения

CASE (с версии 7.40) В команде CASSE (но не WHEN) допустимы арифметические и символьные выражения

CASE TYPE OF ... WHEN TYPE (с версии 7.50)

EQUIV (с версии 7.02) Бинарный инфиксный оператор логической эквивалентность. Все истинные выражения между собой эквиваленты. Все ложные выражения между собой эквивалентны. Ни одно истинное выражение не эквивалентно ни одному истинному

IS INSTANCE OF (с версии 7.50) Унарный постфиксный логический оператор

Логические функции

boolc(*bool* = logical_expression) (с версии 7.02).

Возвращает **строковое** значение длиной 1: «X», если логическое выражение logical_expression истинно, и пробел « » в противном случае

xsdbool(*bool* = logical_expression) (с версии 7.40).

Возвращает **символьное** значение длиной 1: «X», если логическое выражение logical_expression истинно, и пробел « » в противном случае

boolx(*bool* = logical_expression *bit* = bit) (с версии 7.02).

Возвращает шестнадцатиричную строку. Если логическое выражение logical_expression ложно или указано значение bit, равное 0, возвращается строка нулевой длины, если же истинно и задан не нулевой бит, возвращается строка со взведенным битом в позиции, указанной в параметре bit

Предикатные функции (с версии 7.02) и вызовы методов (с версии 7.40 SP8)

contains проверит вхождение строки или результата регулярного выражения

contains_any_of

contains_any_not_of

matches определяет, удовлетворяет ли текст регулярному выражению

Предикатный вызов методов (с версии 7.40 SP8)

Инлайновые декларации (с версии 7.40 SP2)

В командах ABAP, где предполагается получение значения.

DATA(var_name) создает переменные.

FIELD-SYMBOL(<fs_name>) создает филд-символы

```
DATA(first) = 2 + 2.
```

```
READ TABLE tbl ASSIGNING FIELD-SYMBOL(<rec2>) INDEX 1.
```

Операторы конструкторы

NEW	создает данные или объект, результат оператора - ссылка;
VALUE	создает данные, результат оператора - данные;
REF	создает ссылку на (уже имеющиеся) данные или объект;
CONV	о имеющемся данным создает данные указанного типа, то есть осуществляет явное приведение типа данных;
CAST	по имеющейся ссылке на данные или объект создает ссылку на данные или объект указанного типа, то есть осуществляет явное приведение типа ссылки;
EXACT	производит вычисление выражения с максимальной точностью;
COND	условное создание данных, условие – логическое выражение;
SWITCH	условное создание данных, условие – равенство;
CORRESPONDING	разбирает структуры по совпадению имен компонентов.

Ключи внутренних таблиц

Пустой ключ (с версии 7.40) для стандартных таблиц явно указывать отсутствие первичного ключа с помощью фразы WITH EMPTY KEY.

Вторичные ключи (с версии 7.02) Можно определить до 15 вторичных ключей. Вторичные ключи могут быть только сортированные (уникальные и неуникальные) и хэшированные

```
TYPES: BEGIN OF ts
      , first TYPE char12
      , last  TYPE char12
      , passno TYPE char12
      , nick  TYPE char12
      , END   OF ts
      , tt TYPE STANDARD TABLE OF ts
        WITH EMPTY KEY
        WITH NON-UNIQUE SORTED KEY name COMPONENTS last first
        WITH UNIQUE     SORTED KEY nick COMPONENTS nick
        WITH UNIQUE     HASHED KEY  pass COMPONENTS passno.
```

Индексные и ключевые выражения [], функции

```
gs = gt[ 2 ].  
gs = gt[ cos( 0 ) * CONV f( `2.4456` ) ].  
gs = gt[ last = 'БЕССМЕРТНЫЙ' ].  
gs = gt[ KEY pass COMPONENTS passno = '43-03' ].
```

Функция `line_index(...)` возвращает номер записи, удовлетворяющей ключевому выражению. Например

```
line_index( gt[ first = 'КОЩЕЙ' ] )
```

Предикатная функция `line_exists(...)`

Заполнение внутренних таблиц

Команда **MOVE-CORRESPONDING** (с версии 7.40). Копирование по совпадению имен столбцов. По умолчанию все старые записи таблицы-приемника удаляются. Для сохранения старого содержимого **KEEPING TARGET LINES** . Для копирования содержимого внутренних таблиц, компонентами записей которых являются вложенные таблицы можно использовать конструкцию **EXPANDING NESTED TABLES KEEPING TARGET LINES**

Конструкторы **VALUE** и **NEW**.

Фраза **WHERE** позволяет осуществить фильтрацию.

Фраза **FOR** позволяет осуществить проекцию и вычисление значений столбцов.

Фраза **BASE** позволяет добавить содержимое таблицы целиком до вставки, определяемой по фразам **WHERE** и **FOR**.

Итоги по внутренним таблицам. REDUCE

Конструктор REDUCE (с версии 7.40 SP02)

сводит внутреннюю таблицу к элементарному данному или к структуре.

При этом можно задать правила вычисления полей результата. Это удобно для вычисления итогов по таблице. Синтаксис следующий:

```
REDUCE type(  
  INIT var1 exp1i var2 exp2i ...  
  FOR rec  
  IN tbl  
  WHERE ( cond )  
  NEXT var1 exp1n var2 exp2n ...  
).
```

Группировка по внутренней таблице. GROUP BY

```
LOOP LOOP AT tbl INTO rec  
GROUP BY ( fld_grp1 = fld_rec1 ... ) order INTO DATA(grp)
```

```
LOOP LOOP AT tbl INTO rec  
GROUP BY ( fld_grp1 = fld_rec1 ... ) order INTO FIELD-SYMBOL(<grp>)
```

Внутри группировки можно устроить цикл командой

LOOP AT GROUP или подвести итоги суперпозицией конструкторов
VALUE и REDUCE

Ассоциация внутренних таблиц. MESH

Новый составной тип данных.

Задаёт такое составное данное, которое состоит из внутренних таблиц с наложенными между ними ассоциациями.

Позволяет как бы «джойнить» внутренние таблицы

Ассоциация внутренних таблиц. MESH

```
TYPES: BEGIN      OF ts_pet
      / species TYPE char10
      / name    TYPE char20
      / tail    TYPE p LENGTH 4 DECIMALS 2
      / END      OF ts_pet
      / BEGIN    OF ts_prod
      / name    TYPE char20
      / product TYPE char20
      / count   TYPE i
      / END      OF ts_prod
      / tt_pet  TYPE SORTED TABLE OF ts_pet WITH UNIQUE KEY name
      / tt_prod TYPE STANDARD TABLE OF ts_prod WITH EMPTY KEY .
TYPES: BEGIN OF MESH tm_mesh
      / prod TYPE tt_prod ASSOCIATION prod_to_pet TO pet ON name = name
      / pet  TYPE tt_pet
      / END OF MESH tm_mesh.
```

ООП. Повторное исполнение

```
REPORT zqk020102. " Correction and result output
PARAMETERS
: p_first  LENGTH 3 DEFAULT 2
, p_second LENGTH 3 DEFAULT 'oNe'
.
DATA
: gv_result TYPE i
, gx_error  TYPE REF TO cx_root
.

START-OF-SELECTION.
TRY.
    gv_result = p_first + p_second.
    WRITE:/ gv_result.
    CATCH cx_sy_conversion_no_number INTO gx_error.
        CASE p_second.
            WHEN 'ONE'.
                p_second = 1.
                RETRY.
        ENDCASE.
    CATCH cx_root INTO gx_error.
        WRITE: / gx_error->get_text( )
        ,/ cl_abap_classdescr=>describe_by_object_ref( gx_error
        )->ABSOLUTE_NAME.
    ENDTRY.
```

Вебинар Обучение АВАР.
Возврат к обработке после исключений.
RETRY и **RESUME**
19 февраля 2019

<https://www.youtube.com/watch?v=9E0Y07ePXfQ>

ООП. «Возобновляемые» исключения

```
DATA
: gv_result TYPE i
, gv_second type i
, gx_error TYPE REF TO cx_root
.

START-OF-SELECTION.
  TRY.
    TRY.
      gv_second = p_second.
      CATCH cx_root INTO gx_error.
        RAISE RESUMABLE EXCEPTION TYPE cx_sy_conversion_no_number
          EXPORTING previous = gx_error
        .
      FORMAT COLOR COL_GROUP.
    ENDTRY.
    gv_result = p_first + p_second.
    WRITE: / gv result.
    CATCH BEFORE UNWIND cx_sy_conversion_no_number. "
      CASE p_second.
        WHEN 'ONE'.
          p_second = 1.
        ENDCASE.
      RESUME.
    CATCH cx_root INTO gx_error.
      WRITE: / gx_error->get_text( )
        ,/ cl_abap_classdescr=>describe_by_object_ref( gx_error
          )->absolute_name.
    ENDTRY.
```

Вебинар Обучение АВАР.
Возврат к обработке после исключений.
RETRY и **RESUME**
19 февраля 2019

<https://www.youtube.com/watch?v=9E0Y07ePXfQ>

Новый синтаксис Open SQL

Поля во фразах `SELECT`, `GROUP BY`, `ORDER BY` разделяются запятыми, хост-переменные начинаются с `@`

Инлайновая декларация переменной – приемника

`RIGHT [OUTER] JOIN` (с версии 7.4)

`UNION` (с версии 7.5)

Вычисляемые поля. Операции, функции, `CASE`-выражения и `CAST`-выражения в вычисляемых полях

Вебинар Обучение ABAP.
Новое в OpenSQL
25 июля 2018

<https://www.youtube.com/watch?v=eeepK1bV6CY>

Каналы сообщений. Создание приложения и канала

The screenshot shows the SAP SAMC Channels administration interface. At the top, the application is identified as '2QK_WRUAW74' with the description 'A Test'. The 'Channels' tab is active, showing a 'Channel List' on the left with one entry: '/test'. The 'Channel Detail' pane on the right shows the following fields:

- Channel: /test
- Message Type ID: TEXT (highlighted in yellow)
- Activity Scope: Client
- Virus Scan ID (outgoing):

Below these fields is a table titled 'Attributes for Authorized Program' with columns 'Authorized Program', 'Prog. Type', and 'Activity'. Two callout boxes provide additional information:

- A box pointing to the 'TEXT' value in the Message Type ID field contains the text: "TEXT" ABAP type "STRING", "BINARY" ABAP type "XSTRING", "PCP" for Push Channel Protocol: ABAP interface IF_AC_MESSAGE_TYPE_PCP.
- A box pointing to the 'Client' value in the Activity Scope field contains the text: "System", "Client", "User".

SAMC Create
SE80 Create → Connectivity → ABAP Message Channel

Отправка сообщения в канал

```
REPORT ZQK_WRAW74_4_01. " 4.01.Sender
DATA go_sender TYPE REF TO if_amc_message_producer text.
PARAMETERS p_txt TYPE string.
START-OF-SELECTION.
TRY.
  go_sender ?= cl_amc_channel_manager=>create_message_producer(
    i_application_id = 'ZQK_AMC_01'
    i_channel_id      = '/test'
  ).
  go_sender->send( p_txt ).
CATCH cx_root INTO DATA(gx_xcpt).
  DATA(gv_err) = gx_xcpt->get_text( ).
  MESSAGE gv_err TYPE 'I'.
ENDTRY.
WRITE: 'BOT Tak!'.
```

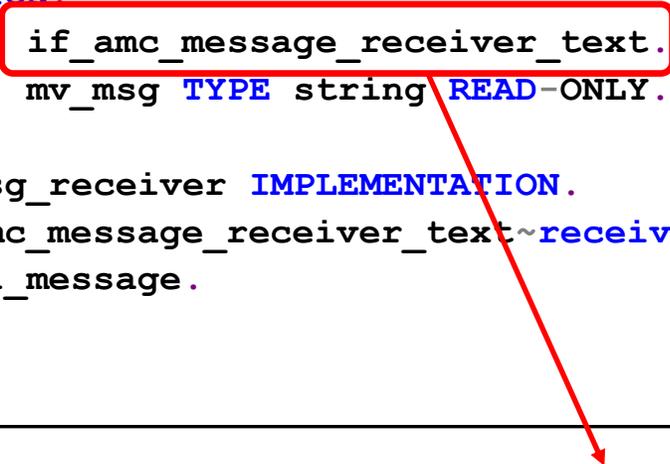
Channel	Message Type ID	Activity Scope	Virus Scan ID (outgoing)
/test	TEXT	Client	

Authorized Program	Prog. Type	Activity
ZQK_WRUAW74_4_01	REPS	Send

- S Send
- R Receive via session
- C Receive via APC WebSocket

Получение сообщения из канала. Класс ресивер

```
CLASS lcl_msg_receiver DEFINITION.  
PUBLIC SECTION  
  INTERFACES if_amc_message_receiver_text.  
  DATA      mv_msg TYPE string READ-ONLY.  
ENDCLASS.  
CLASS lcl_msg_receiver IMPLEMENTATION.  
METHOD if_amc_message_receiver_text~receive.  
  mv_msg = i_message.  
ENDMETHOD.  
ENDCLASS.
```



Тип сообщения	Интерфейс
TEXT	IF_AMC_MESSAGE_RECEIVER_TEXT
BINARY	IF_AMC_MESSAGE_RECEIVER_BYNARY
PCP	IF_AMC_MESSAGE_RECEIVER_PCP

Консумер

```
CREATE OBJECT go_receiver.  
go_consumer = cl_amc_channel_manager=>create_message_consumer(  
    i_application_id = 'ZQK_WRAW74'  
    i_channel_id      = '/test'      ).  
go_consumer->start_message_delivery( go_receiver ).
```

```
WAIT FOR MESSAGING CHANNELS UNTIL  
    go_receiver->  
>mv_msg IS NOT INITIAL UP TO 5 SECONDS.  
IF go_receiver->mv_msg IS NOT INITIAL.  
    MESSAGE go_receiver->mv_msg TYPE 'I'.  
ENDIF.
```

```
go_consumer->stop_message_delivery( go_receiver ).
```

Создаем ресивер,
создаем консумер,
начинаем прием

Ждем сообщения,
Реагируем

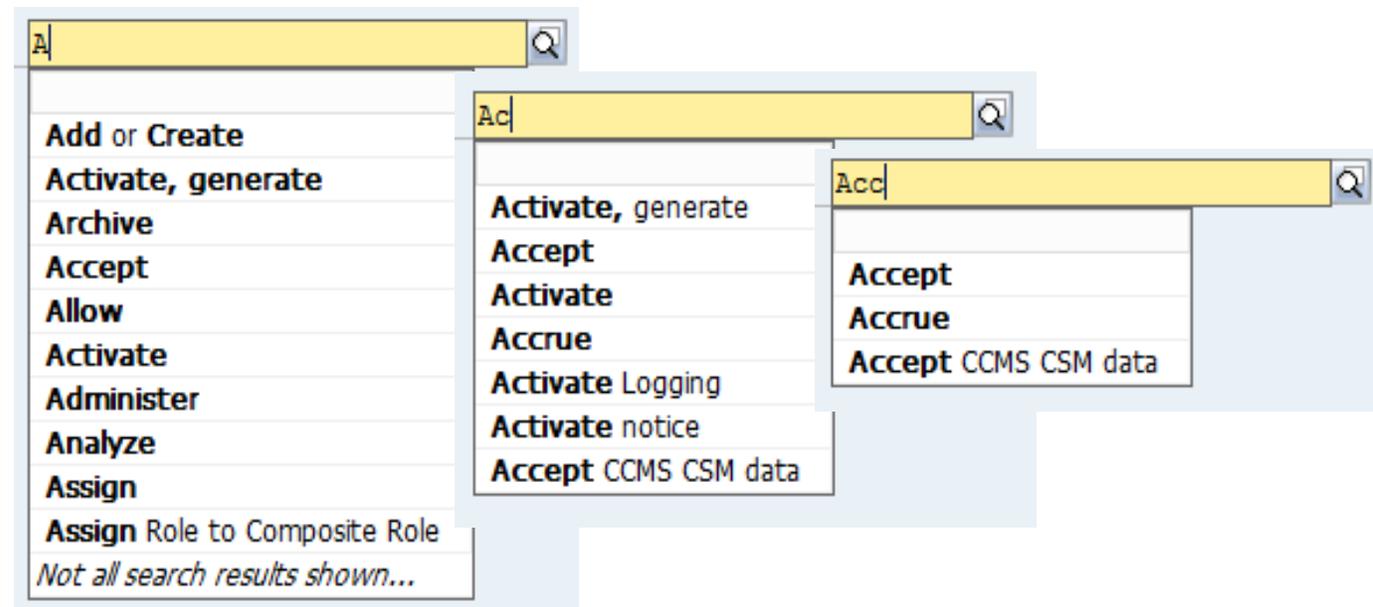
Завершаем прием

АВАР словарь

Баоспецифические свойства таблиц. Пока только поколочное/построчное хранение

Новый тип данных – Глобальные временные таблицы

«Опережающий» search help.



Open SQL и CDS

Structured Query Language (SQL)			
Data Manipulation Language (DML)	Data Definition Language (DDL)	Data Control Language (DCL)	Transaction Control Language (TCL)
SELECT DELETE INSERT UPDATE	CREATE ALTER DROP	GRANT REVOKE DENY	COMMIT ROLLBACK
Есть в ABAP	Нет в ABAP Есть в CDS	Нет в ABAP Есть в CDS	Есть в ABAP

CDS View позволяет сверх того



- определять View ;
- определять профили ;
- использовать функции преобразования единиц измерения и кусов валют.

Ну и вообще это красиво.

Core Data Services: DDL

Возможны вычисляемые поля. Операции, функции (например, конверсия курса валют), **CASE**-выражения и **CAST**-выражения в вычисляемых полях

Возможны внутреннее и внешнее соединение (join) и объединение (union)

Источником могут быть не только таблицы базы данных, но и CDS view

Входные параметры. Ассоциации, открытые ассоциации.

Большой набор аннотаций

```
define view zqk2_03_B as
  select from scarr
  { mandt
  , carrid
  , carrname
  , currcode
  , url
  }
```

Вебинар Обучение ABAP.
Простые CDS
24 октября февраля 2018

<https://www.youtube.com/watch?v=eepK1bV6CY>

Core Data Services: DCL

Привилегии задаются только на CDS view, не на таблицы базы данных !

При обращении к **ZQK431** интерфейс к базе данных автоматически добавит к WHERE-фазе соответствующие условия фильтрации

```
define role Zqk451 {  
    grant select on ZQK431  
        where carrid <> 'LH'  
        and carrid <> 'AA';  
}
```

```
define role Zqk451 {  
    grant select on ZQK431  
        where ( carrid ) = aspect pfcg_auth( s_carrid,  
        carrid, actvt = '03' );  
}
```

Core Data Services: расширения

При обращении к расширяемому view к нему будут добавлены поля из расширений

```
extend view ZQK431  
with ZQK432  
{ spfli.airpfrom  
, spfli.airpto  
}
```

В каких семинарах говорится о доступных возможностях SQL в ABAP

Платформенно-независимые возможности

Подмножества SQL	Реализовано		
	ABAP Open SQL	ABAP Словарь	CDS
DML	BC400 , BC402 , BC414 , BC490		
DDL		BC430	HA300 , HA400 , S4D430
DCL			S4D430
TCL	BC414		

Платформенно-зависимые возможности

SQL в полном объеме	EXEC SQL. ... ENDEXEC.	
	ABAP Database Connectivity (ADBC)	HA400
	ABAP Managed Database Procedures (AMDP)	

Семинар **D75AW**. Дельта-курс: с ABAP Workbench SAP NetWeaver 7.0 на SAP NetWeaver 7.5

Дельта-курс: с ABAP
Workbench SAP
NetWeaver 7.0 на SAP
NetWeaver 7.5

D75AW

3 дн.

[Официальное описание](#)

[Дополнительное описание](#)



- Разработчики
- Консультанты в области разработки



- Введение
- Окружение
- Выражения в ABAP
- Возможности внутренних таблиц
- Изменения в Open SQL
- Core Data Services (CDS)
- Каналы сообщений ABAP
- Новые инструменты анализа



Требования

Обязательно

Опыт процедурно- и объектно-ориентированной разработки на ABAP в SAP NetWeaver 7.0

Приобретайте знания вместе с SAP Training and Adoption



Страница SAP Training and Adoption
<https://www.sap.com/cis/training-certification.html>



Актуальные новости
<https://training.sap.com/content/CIS-RUNews>



Электронный магазин семинаров
www.training.sap.com



[@SAPtvCIS](#)



[@SAPEducation](#)

Москва, Учебный центр SAP
+7 (495) 797 27 20
education.russia@sap.com