# Система интеллектуальной обработки команд сканера MINDEO MD6600AT-HD для ассоциирования кассет, контейнеров, пластин полупроводникового производства

ПОСТАНОВКА ЗАДАЧИ:

В процессе обработки партий пластин в полупроводниковом производстве необходимо обеспечить идентификацию и прослеживаемость продукции на всех этапах производства. На этапе формирования партий происходит отмывка кассет и контейнеров заданной протокольной зоны и перемещение пластин в кассету. Физически партия пластин находится в кассете, кассета находится в контейнере. Кассеты, контейнеры используются только в пределах своей протокольной зоны. Кассеты физически не связаны с контейнерами и могут быть помещены в другой контейнер в пределах протокольной зоны. Каждая партия, кассета, контейнер имеет идентификационную карту со штрих-кодом и цветным признаком определенной протокольной зоны: FEOL/MOL/BEOL.

В производственном маршруте может быть до тысячи технологических операций. Партия на определенных этапах маршрута переходит в другую протокольную зону, с перемещением пластин из «старой» кассеты/контейнера в «новую» кассету/контейнер заданной протокольной зоны.

Для однозначного соответствия требуется иметь возможность логически связывать кассету, контейнер (ассоциировать партию/кассету/контейнер), партию и разрывать их соответствие после окончания обработки партии в заданной протокольной зоне (диссоциировать партию/кассету/контейнер).

Нужно с помощью barcode сканера периодически проверять, что партия/кассета/контейнер логически связаны с друг другом и соответствуют протокольной зоне и ассоциировать/диссоциировать экземпляры партий, кассет, контейнеров.

Для решения этой задачи был разработан драйвер сканера MINDEO MD6600AT-HD, который обеспечивает вызов хранимой процедуры mindeo\_command с входным параметром @CommandString nvarchar(1024) = NULL

В стандартном режиме драйвер сканера транслирует коды команд в текущем поле ввода данных.

При сканировании набора символов, заканчивающего на символ '$' происходит перевод сканера в режим интеллектуальной обработки команд. При сканировании команд выполняется операционный сценарий, который может содержать несколько параметров. Для возврата в режим стандартного сканирования используется специальная команда.

При сканирование с драйвера приходит входящая строка символов

@CommandString nvarchar(1024) = NULL

Операнды команды разделены символами ';'

Каждое новое сканирование сканера к существующей последовательности символов добавляет операнд с разделителем ';'

Первый операнд - IP-адрес компьютера, к которому подключен сканер MINDEO MD6600AT-HD

Второй операнд - Имя компьютера, к которому подключен сканер MINDEO MD6600AT-HD

Третий операнд - - команда операционного сценария, для выполнения заданных действий

Следующие 6 операндов - параметры операционного сценария, для выполнения заданных действий.

В следующем примере рассмотрен следующий операционный сценарий:

* Сканирование сканером штрих-кода команды
* Сканирование одного или нескольких штрих-кодов этикеток объектов
* Выполнение заданных действий, в соответствии с операционным сценарием

Например, при последовательном сканировании 4 объектов приходят строки:

1. '192.168.1.101;DT-AMORENKO;AW2000T0000$;'
2. '192.168.1.101;DT-AMORENKO;AW2000T0000$;E712005;'
3. '192.168.1.101;DT-AMORENKO;AW2000T0000$;E712005;M01;'
4. '192.168.1.101;DT-AMORENKO;AW2000T0000$;E712005;M01;BOPM01;'

* Сканирование метки команды АССОЦИАЦИИ ПАРТИЯ/КАССЕТА/КОНТЕЙНЕР
* Сканирование метки партии, кассеты, контейнера (в любой последовательности)
* Отображение результатов проверки соответствия.
* Коррекция таблиц KARAT\_LABELS и БД MES E\_CS\_MFDUNIT\_DETAILS, E\_CS\_MFDUNIT\_CUSTOM, E\_CS\_OBJECT\_DETAILS, E\_CS\_VARIABLE\_DETAILS
* Посредством триггера trgUpd\_KARAT\_LABELS в таблице KARAT\_LABELS\_HISTORY сохраняется история изменений KARAT\_LABELS и в таблице KARAT\_LABELS\_LOG ведется журнал выполненных команд

В результате обработки хранимой процедуры драйверу возвращаются параметры

SELECT

@CommandString AS CommandString, -- Входящая строка символов

@MessageHeader AS MessageHeader, -- Заголовок Исходящего сообщения

@MessageText AS MessageText, -- Текст Исходящего сообщения

@MessageColor AS MessageColor, -- RBG Цвет символов Исходящего сообщения

@MessageBgColor AS MessageBgColor,-- RBG Цвет фона Исходящего сообщения

@MessageTimeout AS MessageTimeout,-- Длительность Исходящего сообщения

@MessageSound AS MessageSound, -- Тип Звука исходящего сообщения

@MessageStop AS MessageStop, -- Признак остановки операционного сценария и

-- переход в режим стандартного сканирования

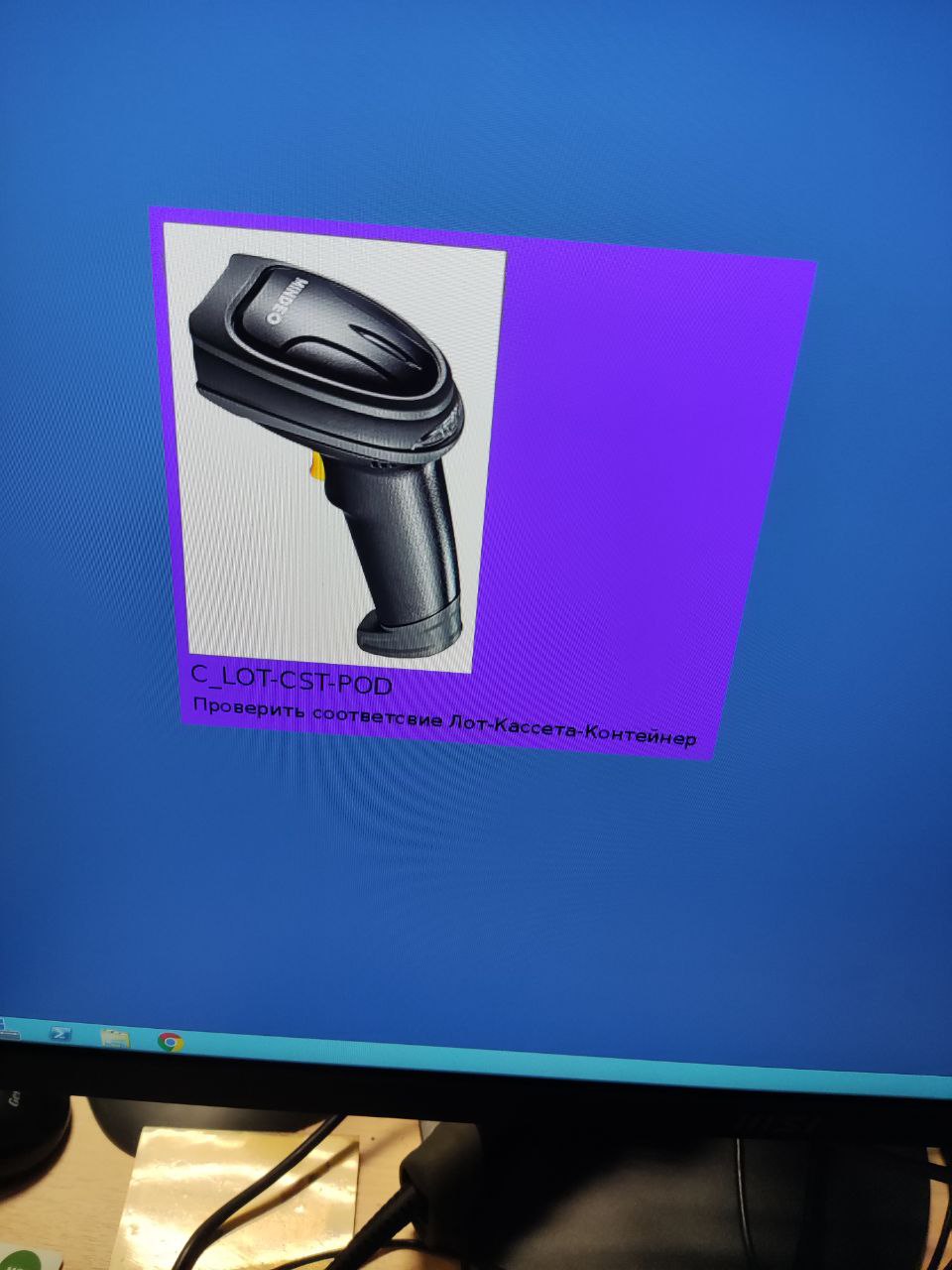
@MessageInject AS MessageInject; -- Ручной ввод данных, если необходимо

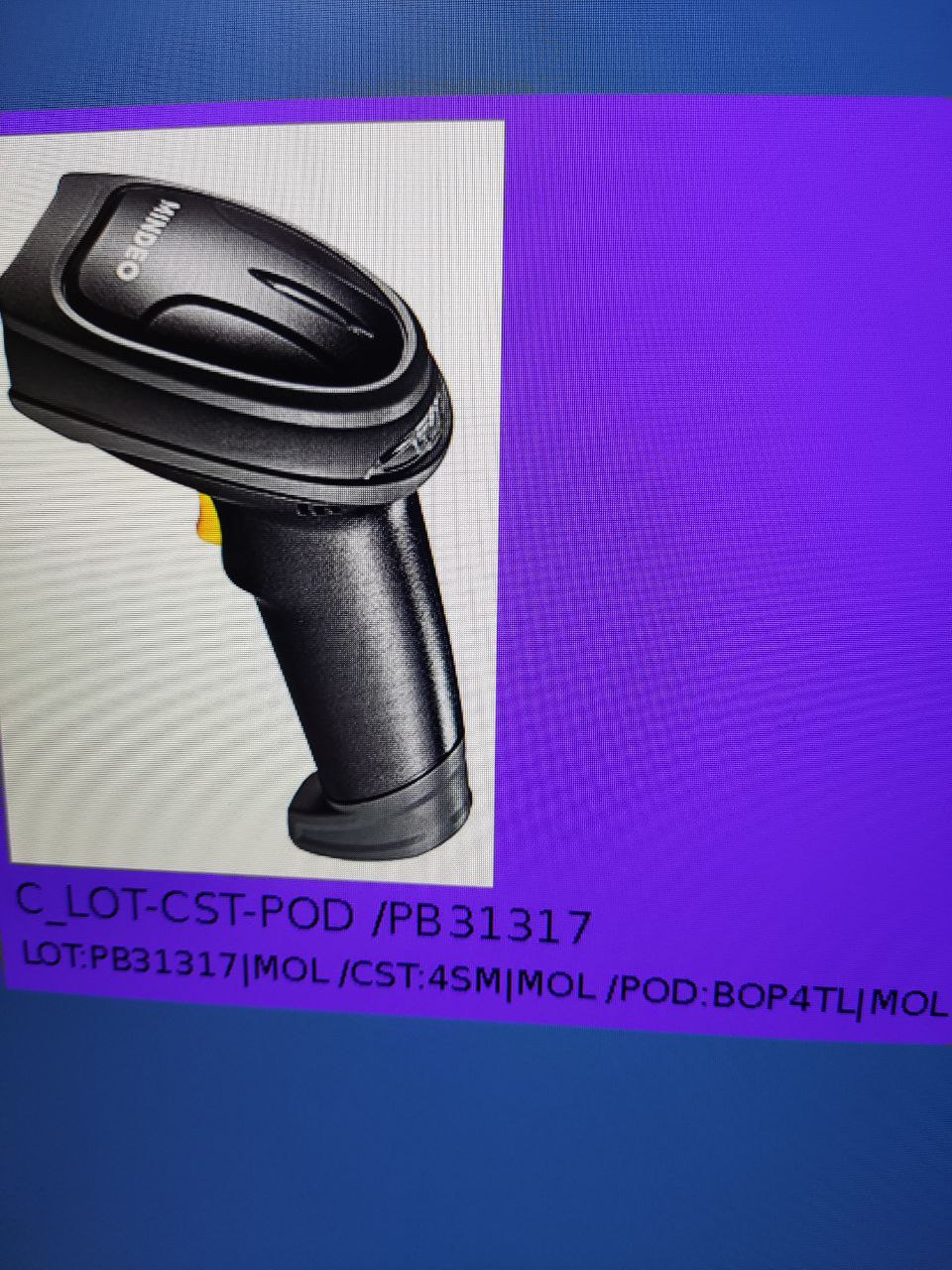
Возврат драйверу сканера результатов выполнения каждой команды сценария после каждого сканирования в виде окна сообщений с соответствующим звуковым сигналом.

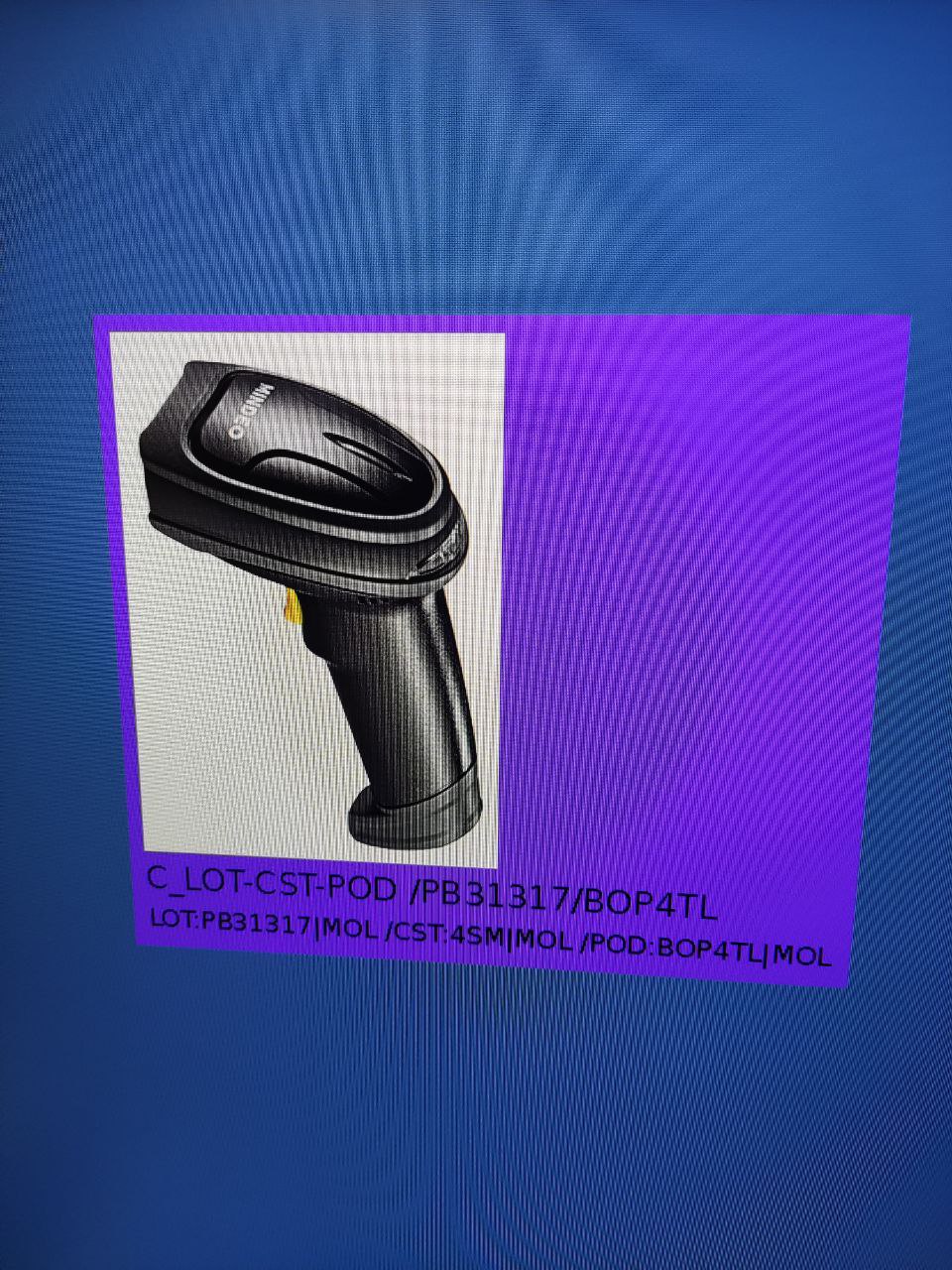
Источником данных является MES система, для ввода кассет/контейнеров используется

Специальная комманда

ПРИМЕР СРИНШОТОВ СКАНИРОВАНИЯ







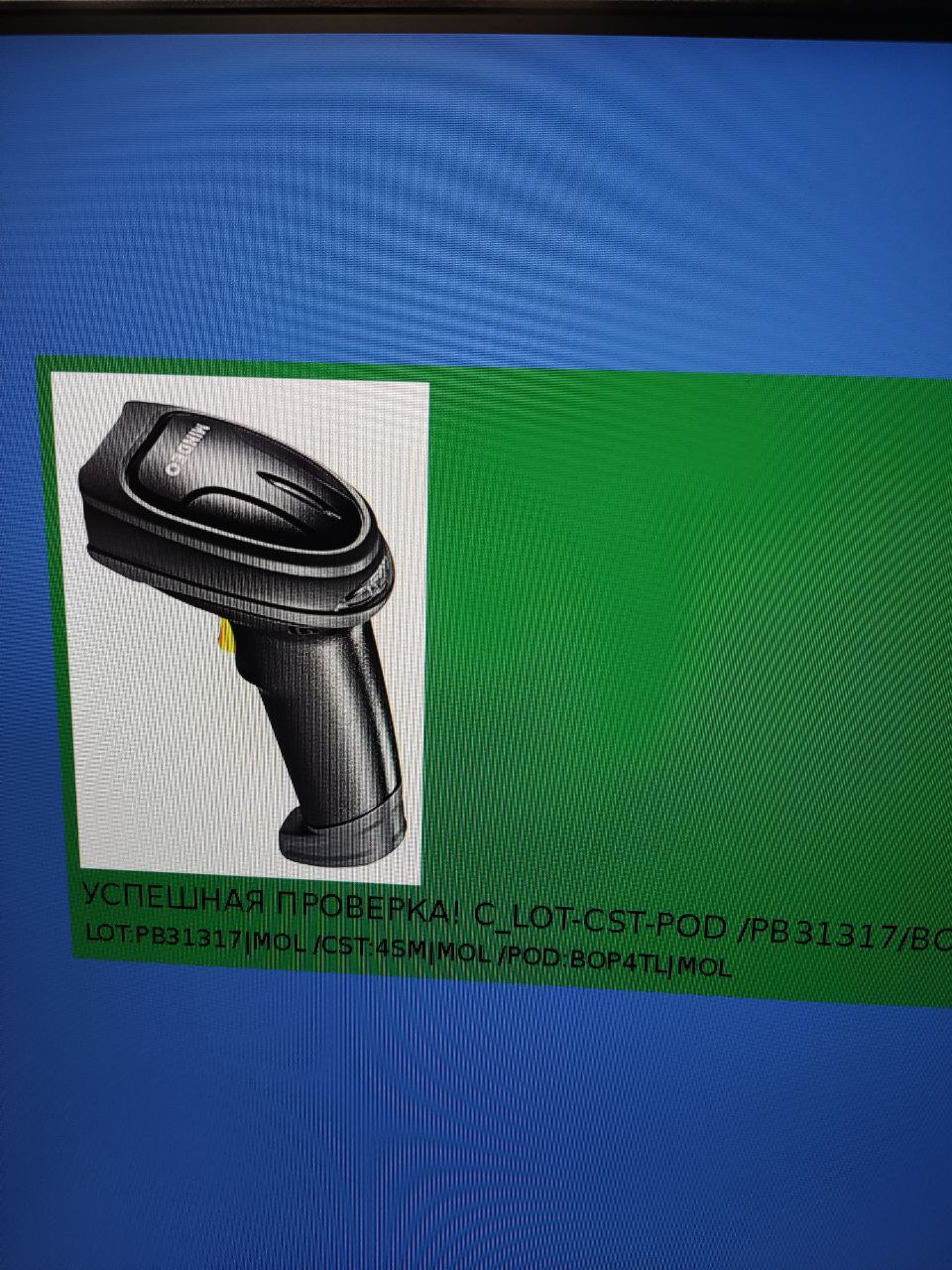
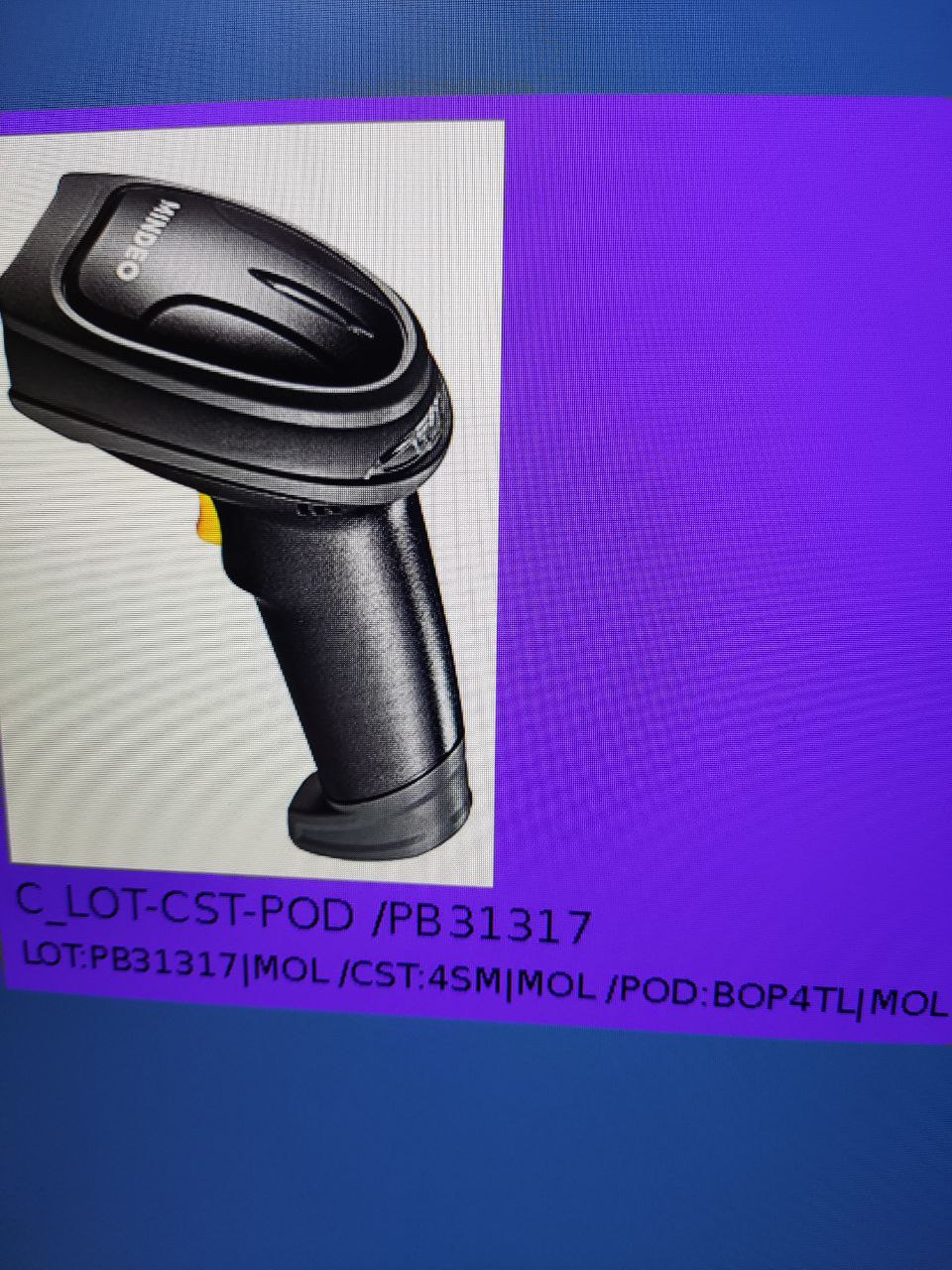
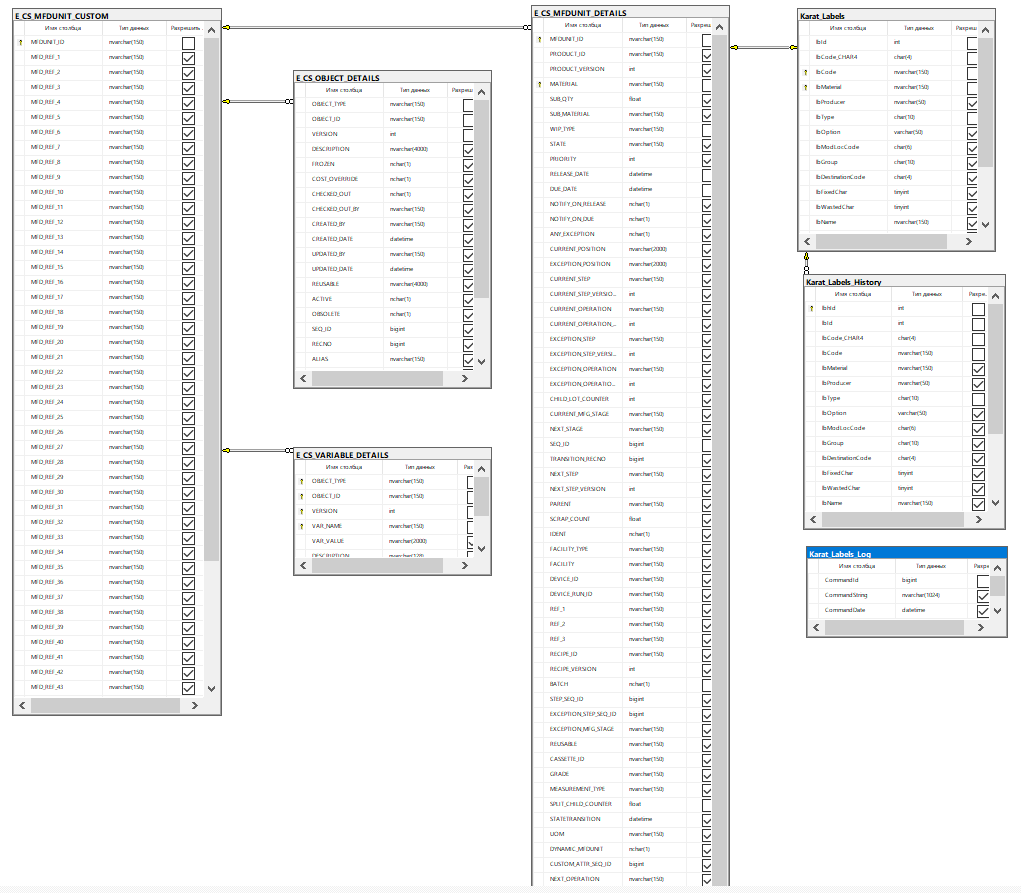


СХЕМА ДАННЫХ

U

USE [KARAT\_TEST]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[mindeo\_command] Script Date: 11.12.2024 19:34:54 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ============================================

ALTER PROCEDURE [dbo].[mindeo\_command]

(

-- Add the parameters for the stored procedure here

@CommandString nvarchar(1024) = NULL

)

AS

BEGIN

-- ===============================

-- SET NOCOUNT added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- ===============================

-- Procedure variables

DECLARE @Command nvarchar(64)=NULL,

@Param1 nvarchar(64)=NULL,

@Param2 nvarchar(64)=NULL,

@Param3 nvarchar(64)=NULL,

@Param4 nvarchar(64)=NULL,

@Param5 nvarchar(64)=NULL,

@Param6 nvarchar(64)=NULL,

-- So many as you want ...

@Terminator nvarchar(64)=NULL,

-- temp var

@Command\_temp nvarchar(64)=NULL,

@Command\_counter int = 0,

@CompIP nvarchar(32)=NULL,

@CompNAME nvarchar(32)=NULL,

@CommandNAME varchar(64)=NULL,

@UserNAME varchar(64)=NULL,

@Param\_counter int = 0;

-- ===============================

-- main cursor

DECLARE command\_cursor CURSOR FOR

SELECT value FROM STRING\_SPLIT(@CommandString, ';');

-- ===============================

OPEN command\_cursor;

-- -------------------------------

FETCH NEXT FROM command\_cursor

INTO @Command\_temp

-- -------------------------------

WHILE @@FETCH\_STATUS = 0

-- -------------------------------

BEGIN

-- -------------------------------

IF @Command\_counter = 0 SET @CompIP = @Command\_temp

ELSE IF @Command\_counter = 1 SET @CompNAME = @Command\_temp

ELSE IF @Command\_counter = 2 SET @Command = @Command\_temp

ELSE IF @Command\_counter = 3 SET @Param1 = @Command\_temp

ELSE IF @Command\_counter = 4 SET @Param2 = @Command\_temp

ELSE IF @Command\_counter = 5 SET @Param3 = @Command\_temp

ELSE IF @Command\_counter = 6 SET @Param4 = @Command\_temp

ELSE IF @Command\_counter = 7 SET @Param5 = @Command\_temp

ELSE IF @Command\_counter = 8 SET @Param6 = @Command\_temp

-- -------------------------------

SET @Command\_counter += 1;

SET @Terminator = @Command\_temp

-- -------------------------------

FETCH NEXT FROM command\_cursor

INTO @Command\_temp

-- -------------------------------

END

-- -------------------------------

CLOSE command\_cursor;

DEALLOCATE command\_cursor;

SET @Param\_counter = @Command\_counter - 4;

-- ===============================

DECLARE @MessageHeader nvarchar(64) = N'ERROR!', -- Header

@MessageText nvarchar(1024) = N'', -- Text

@MessageColor int = CONVERT (int, 0xE000E), -- RBG Text COLOR.

@MessageBgColor int = CONVERT (int, 0x770FF0), -- RBG Back ground COLOR..

@MessageTimeout float = 10.5, -- Time in sec message preview on screen

@MessageSound int = 3, -- Sound file play

@MessageStop int = 0,

@MessageInject nvarchar(1024) = ''--- N'29,56,49,49,56,29,0,0,0,0,42,35,35,42,18,18,38,38,38,38,24,24,57,57,42,17,17,42,24,24,19,19,38,38,32,32,57,57,42,2,2,42'; -- Inject KEY\_CODE sequence

SET @Command\_temp='';

IF @Command = 'AW1000S0000$' BEGIN

-- -------------------------------

SET @CommandName='SCANER\_MODE'

SET @MessageHeader = @CommandName;

SET @MessageTimeout =5;

SET @MessageBgColor=CONVERT (int, 0x008050);

SET @MessageSound =2;

SET @MessageStop =1;

SET @MessageInject ='';

SET @MessageText = N'Включен режим сканирования без команд';

-- -------------------------------

END ELSE

-- ===============================

IF @Command = 'AW2000T0000$' BEGIN

-- -------------------------------

SET @CommandName='A\_LOT-CST-POD'

SET @MessageHeader = @CommandName ;

SET @MessageTimeout =5;

SET @MessageSound =0;

SET @MessageStop =0;

SET @MessageText= N'Ассоциировать Лот-Кассета-Контейнер '

if @Command\_counter = 3 SET @MessageText = ISNULL(RTRIM(@MessageText),'') + N' ' + ISNULL(LTRIM(@Param1),'')--+@CommandString

IF @Command\_counter > 3

EXEC [Karat\_TEST].[dbo].[mindeo\_command\_Execute]

@CommandName=@CommandName, @Param\_counter=@Param\_counter, @CompIP=@CompIP, @CompName=@CompName, @UserName=@CompName,

@Param1 = @Param1, @Param2 = @Param2, @Param3 = @Param3,

@Param4 = @Param4, @Param5 = @Param5, @Param6 = @Param6,

@CommandString = @CommandString,

@MessageHeader = @MessageHeader OUTPUT,

@MessageText = @MessageText OUTPUT,

@MessageColor = @MessageColor OUTPUT,

@MessageBgColor = @MessageBgColor OUTPUT,

@MessageTimeout = @MessageTimeout OUTPUT,

@MessageSound = @MessageSound OUTPUT,

@MessageStop = @MessageStop OUTPUT,

@MessageInject = @MessageInject OUTPUT

-- -------------------------------

END ELSE

IF @Command = 'AW3000T0000$' BEGIN

-- -------------------------------

SET @CommandName='C\_LOT-CST-POD'

SET @MessageHeader = @CommandName;

SET @MessageTimeout =5;

SET @MessageSound =0;

SET @MessageStop =0;

SET @MessageText= N'Проверить соответсвие Лот-Кассета-Контейнер '

if @Command\_counter = 3 SET @MessageText = ISNULL(RTRIM(@MessageText),'') + N' ' + ISNULL(LTRIM(@Param1),'');

IF @Command\_counter > 3 EXEC [Karat\_TEST].[dbo].[mindeo\_command\_Execute]

@CommandName=@CommandName, @Param\_counter=@Param\_counter, @CompIP=@CompIP, @CompName=@CompName, @UserName=@CompName,

@Param1 = @Param1, @Param2 = @Param2, @Param3 = @Param3,

@Param4 = @Param4, @Param5 = @Param5, @Param6 = @Param6,

@CommandString = @CommandString,

@MessageHeader = @MessageHeader OUTPUT,

@MessageText = @MessageText OUTPUT,

@MessageColor = @MessageColor OUTPUT,

@MessageBgColor = @MessageBgColor OUTPUT,

@MessageTimeout = @MessageTimeout OUTPUT,

@MessageSound = @MessageSound OUTPUT,

@MessageStop = @MessageStop OUTPUT,

@MessageInject = @MessageInject OUTPUT

-- -------------------------------

END ELSE

IF @Command = 'AW4000T0000$' BEGIN

-- -------------------------------

SET @CommandName='D\_LOT-CST-POD';

SET @MessageHeader = @CommandName;

SET @MessageTimeout =5;

SET @MessageSound =0;

SET @MessageStop =0;

SET @MessageText = N'Диссоциировать Лот-Кассета-Контейнер';

if @Command\_counter = 3 SET @MessageText = ISNULL(RTRIM(@MessageText),'') + N' ' + ISNULL(LTRIM(@Param1),'');

IF @Command\_counter > 3

EXEC [Karat\_TEST].[dbo].[mindeo\_command\_Execute]

@CommandName=@CommandName, @Param\_counter=@Param\_counter, @CompIP=@CompIP, @CompName=@CompName, @UserName=@CompName,

@Param1 = @Param1, @Param2 = @Param2, @Param3 = @Param3,

@Param4 = @Param4, @Param5 = @Param5, @Param6 = @Param6,

@CommandString = @CommandString,

@MessageHeader = @MessageHeader OUTPUT,

@MessageText = @MessageText OUTPUT,

@MessageColor = @MessageColor OUTPUT,

@MessageBgColor = @MessageBgColor OUTPUT,

@MessageTimeout = @MessageTimeout OUTPUT,

@MessageSound = @MessageSound OUTPUT,

@MessageStop = @MessageStop OUTPUT,

@MessageInject = @MessageInject OUTPUT

-- -------------------------------

END ELSE

IF @Command = 'AW0000T0000$' BEGIN

-- -------------------------------

SET @CommandName='I\_LOT-CST-POD'

SET @MessageHeader = @CommandName;

SET @MessageTimeout =5;

SET @MessageSound =0;

SET @MessageStop =0;

SET @MessageText= N'Ввод новых Лот-Кассета-Контейнер '

if @Command\_counter = 3 SET @MessageText = ISNULL(RTRIM(@MessageText),'') + N' ' + ISNULL(LTRIM(@Param1),'')--+@CommandString

IF @Command\_counter > 3

EXEC [Karat\_TEST].[dbo].[mindeo\_command\_Execute]

@CommandName=@CommandName, @Param\_counter=@Param\_counter, @CompIP=@CompIP, @CompName=@CompName, @UserName=@CompName,

@Param1 = @Param1, @Param2 = @Param2, @Param3 = @Param3,

@Param4 = @Param4, @Param5 = @Param5, @Param6 = @Param6,

@CommandString = @CommandString,

@MessageHeader = @MessageHeader OUTPUT,

@MessageText = @MessageText OUTPUT,

@MessageColor = @MessageColor OUTPUT,

@MessageBgColor = @MessageBgColor OUTPUT,

@MessageTimeout = @MessageTimeout OUTPUT,

@MessageSound = @MessageSound OUTPUT,

@MessageStop = @MessageStop OUTPUT,

@MessageInject = @MessageInject OUTPUT

-- -------------------------------

END ELSE

BEGIN

-- -------------------------------

SET @MessageHeader =N'ERROR';

SET @MessageText= N'Неизвестная комманда '+@CommandString;

SET @MessageTimeout =5;

SET @MessageSound =1;

SET @MessageStop =1;

END

IF @CommandString LIKE '%$%' AND @CommandString IS NOT NULL BEGIN

-- Сохранить историю команд сканера. Команда содерижит '$'

-- ОСТАЛЬНОЕ ИГНОРИРОВАТЬ

BEGIN TRAN

INSERT INTO KARAT\_TEST.dbo.Karat\_labels\_Log (CommandString) VALUES (@CommandString);

IF @@ERROR<0

BEGIN

SET @MessageHeader=N'ОШИБКА записи команды! '+ ISNULL(@MessageHeader, '');

ROLLBACK TRAN

END

COMMIT TRAN

END;

-- ===============================

-- Out procedure result

-- Out procedure result

--SET @MessageText = @CommandString

SELECT @CommandString AS CommandString,

@MessageHeader AS MessageHeader,

@MessageText AS MessageText,

@MessageColor AS MessageColor,

@MessageBgColor AS MessageBgColor,

@MessageTimeout AS MessageTimeout,

@MessageSound AS MessageSound,

@MessageStop AS MessageStop,

@MessageInject AS MessageInject;

-- ===============================

END

USE [KARAT\_TEST]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[mindeo\_command\_Execute] Script Date: 11.12.2024 19:35:17 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[mindeo\_command\_Execute]

(

-- Add the parameters for the stored procedure here

@CommandName nvarchar(64) = NULL,

@Param\_counter int = 0,

@CompIP nvarchar(32)=NULL,

@CompNAME nvarchar(32)=NULL,

@USERName nvarchar(32)=NULL,

@Param1 nvarchar(64) = NULL, @Param2 nvarchar(64) = NULL, @Param3 nvarchar(64) = NULL,

@Param4 nvarchar(64) = NULL, @Param5 nvarchar(64) = NULL, @Param6 nvarchar(64) = NULL,

@CommandString nvarchar(1024) = NULL,

@MessageHeader nvarchar(64) OUTPUT,

@MessageText nvarchar(1024) OUTPUT,

@MessageColor int OUTPUT,

@MessageBgColor int OUTPUT,

@MessageTimeout float OUTPUT,

@MessageSound int OUTPUT,

@MessageStop int OUTPUT,

@MessageInject nvarchar(1024) OUTPUT

)

AS

BEGIN

DECLARE @N int=0, @IsCheck int=0;

DECLARE @ParamCurrent nvarchar(64)='';

DECLARE @LotId nvarchar(64)=NULL, @CSTId nvarchar(64)=NULL, @PODId nvarchar(64)=NULL;

DECLARE @LotProtocol nvarchar(64)=NULL, @CSTProtocol nvarchar(64)=NULL, @PODProtocol nvarchar(64)=NULL

,@LbType nvarchar(64)=NULL, @LbCode nvarchar(64)=NULL;

DECLARE

@lbCodeLOT nvarchar(64)=NULL--, @CodeLOT nvarchar(64)=NULL

,@lbCodeCST nvarchar(64)=NULL--, @CodeCST nvarchar(64)=NULL

,@lbCodePOD nvarchar(64)=NULL--, @CodePOD nvarchar(64)=NULL

,@lbProtocolLOT nvarchar(64)=NULL, @ProtocolLOT nvarchar(64)=NULL, @LbMaterial nvarchar(64)=N'КОД', @Qty int=0

,@lbProtocolCST nvarchar(64)=NULL

,@lbProtocolPOD nvarchar(64)=NULL

,@lbFact\_PlaceLOT nvarchar(64)=NULL

,@lbFact\_PlaceCST nvarchar(64)=NULL

,@lbFact\_PlacePOD nvarchar(64)=NULL

,@ClbCode nvarchar(64)=NULL

,@ClbType nvarchar(64)=NULL

,@CProtocol nvarchar(64)=NULL;

IF @Param\_counter <1 RETURN

IF @Param\_counter = 1 SET @ParamCurrent=@Param1

ELSE IF @Param\_counter = 2 SET @ParamCurrent=@Param2

ELSE IF @Param\_counter = 3 SET @ParamCurrent=@Param3

ELSE IF @Param\_counter =4 SET @ParamCurrent=@Param4

ELSE IF @Param\_counter = 5 SET @ParamCurrent=@Param5

ELSE IF @Param\_counter = 6 SET @ParamCurrent=@Param6

ELSE

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader= N'МНОГО ПАРАМЕТРОВ! '+ ISNULL(@MessageHeader, '');

RETURN

END

IF @CommandName IN ('A\_LOT-CST-POD', 'C\_LOT-CST-POD', 'D\_LOT-CST-POD') AND NOT EXISTS(SELECT \* FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE ClbCode=@ParamCurrent)

BEGIN

-- Неизвестный код

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader=N'ОШИБКА КОДА! '+ISNULL(RTRIM(@MessageHeader),'');

RETURN

END

SELECT TOP (1) @lbCodeLOT=[lbCodeLOT]

,@lbCodeCST=[lbCodeCST]

,@lbCodePOD=[lbCodePOD]

,@lbProtocolLOT=[lbProtocolLOT]

,@lbProtocolCST=[lbProtocolCST]

,@lbProtocolPOD=[lbProtocolPOD]

,@lbFact\_PlaceLOT=[lbFact\_PlaceLOT]

,@lbFact\_PlaceCST=[lbFact\_PlaceCST]

,@lbFact\_PlacePOD=[lbFact\_PlacePOD]

,@ClbCode=[ClbCode]

,@ClbType=[ClbType]

,@CProtocol=[CProtocol]

FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE ClbCode=@ParamCurrent

SELECT TOP (1) @lbCodeLOT=[lbCodeLOT], @lbProtocolLOT=[lbProtocolLOT], @ProtocolLOT=[lbProtocolLOT]

FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE [ClbType]='LOT' AND [ClbCode] IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

SELECT TOP (1) @lbCodeCST=[lbCodeCST], @lbProtocolCST=[lbProtocolCST]

FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE [ClbType]='CST' AND [ClbCode] IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

SELECT TOP (1) @lbCodePOD=[lbCodePOD], @lbProtocolPOD=[lbProtocolPOD]

FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE [ClbType]='POD' AND [ClbCode] IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

IF @CommandName IN ('A\_LOT-CST-POD', 'C\_LOT-CST-POD', 'D\_LOT-CST-POD') BEGIN

-- Предварительная проверка ассоциаций

SET @MessageHeader =ISNULL(RTRIM(@MessageHeader),'')+' '+ISNULL('/'+ RTRIM(@Param1),'')+ISNULL('/'+ RTRIM(@Param2),'')

+ISNULL('/'+ RTRIM(@Param3),'')+ISNULL('/'+ RTRIM(@Param4),'')+ISNULL('/'+ RTRIM(@Param5),'')+ISNULL('/'+ RTRIM(@Param6),'')

SET @MessageText = +ISNULL(N'LOT:'+RTRIM(@lbCodeLOT),'') +'|'+ISNULL(RTRIM(@lbProtocolLOT),'')

+ISNULL(N' /CST:'+ RTRIM(@lbCodeCST),'') +'|'+ISNULL(RTRIM(@lbProtocolCST),'')+ISNULL(' /POD:'+ RTRIM(@lbCodePOD),'') +'|'+ISNULL(RTRIM(@lbProtocolPOD),'');

IF 1<(SELECT TOP 1 COUNT(DISTINCT CONCAT(ISNULL(lbProtocolLOT,''), ISNULL(lbProtocolCST,''), ISNULL(lbProtocolPOD,'')))

FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] where clbcode in (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6))

BEGIN

SET @MessageTimeout =30

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageHeader=N'РАЗНЫЕ ПРОТОКОЛЫ! '+ISNULL(RTRIM(@MessageHeader),'')

IF @CommandName IN ('C\_LOT-CST-POD') RETURN

END

IF 0<

(SELECT IIF(COUNT(lbCodeLOT)>1,1,0)+IIF(COUNT(lbCodeCST)>1,1,0)+IIF(COUNT(lbCodePOD)>1,1,0) FROM

(SELECT DISTINCT lbCodeLOT, lbCodeCST, lbCodePOD FROM [Karat\_TEST].[dbo].[Karat\_VIEW\_Labels] WHERE clbcode IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6) ) a

)

BEGIN

-- Неассоциировать, если была ассоциация нескольких партий, кассет, контейнеров

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader=N'ОШИБКА МНОЖЕСТВЕННОЙ ACCOЦИАЦИИ! '+ISNULL(RTRIM(@MessageHeader),'');

RETURN

END

ELSE BEGIN

IF ISNULL(@lbCodeLOT, '') IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

AND ISNULL(@lbCodeCST, '') IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

AND ISNULL(@lbCodePOD, '') IN (@Param1, @Param2, @Param3, @Param4, @Param5, @Param6)

AND @lbCodeCST IS NOT NULL AND @lbCodePOD IS NOT NULL BEGIN

SET @MessageTimeout =10;

SET @MessageSound =2;

SET @MessageStop = 1;

SET @IsCheck =1;

SET @MessageBgColor=CONVERT (int, 0x008050);

IF @CommandName = 'C\_LOT-CST-POD' SET @MessageHeader=N'УСПЕШНАЯ ПРОВЕРКА! '+ISNULL(RTRIM(@MessageHeader),'');

END BEGIN

-- продолжать

SET @MessageTimeout =5;

SET @MessageStop = 0;

END

END

END -- @CommandName IN ('A\_LOT-CST-POD', 'C\_LOT-CST-POD', 'D\_LOT-CST-POD')

IF @CommandName='A\_LOT-CST-POD' AND @Param\_counter>=3 BEGIN

-- после проверки @CommandName='C\_LOT-CST-POD'

SET @MessageTimeout =5

SET @LotId =ISNULL(dbo.Karat\_ER\_Alias\_TO\_MFDUNIT\_ID(@lbCodeLOT),''); -- получаем MFDUNIT\_ID

SET @PODId =ISNULL(@lbCodePOD,''); SET @CSTId =ISNULL(@lbCodeCST,'');

SET @CSTProtocol =@lbProtocolCST; SET @CSTProtocol =@lbProtocolPOD;

IF @LotId ='' BEGIN

SET @MessageTimeout =30

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageHeader=N'НЕЗАРЕГИСТРИРОВАННАЯ ПАРТИЯ! '+ISNULL(RTRIM(@MessageHeader),'');

RETURN

END

IF @CSTProtocol=@CSTProtocol BEGIN

-- Проверка пройдена протоколы кассеты и контейнера совпадают

-- Ассоциировать партию/кассету/контейнер

SET @LotProtocol =concat(N'LOT\_',@CSTProtocol); -- протокол партии установить как у кассеты

END

ELSE BEGIN

-- Разные протоколы

SET @MessageTimeout =30

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageHeader=N'ОШИБКА ПРОТОКОЛА! '+ISNULL(RTRIM(@MessageHeader),'');

RETURN

END

BEGIN TRAN

UPDATE [Karat\_TEST].[dbo].E\_CS\_VARIABLE\_DETAILS

SET

[VAR\_VALUE] = LTRIM(RTRIM(@CSTId))

WHERE [OBJECT\_TYPE] = 'MFDUNIT' AND [OBJECT\_ID] = @LotId AND [VAR\_NAME] = '$CASSETTE';

UPDATE [Karat\_TEST].[dbo].E\_CS\_VARIABLE\_DETAILS

SET

[VAR\_VALUE] = LTRIM(RTRIM(@PODId))

WHERE [OBJECT\_TYPE] = 'MFDUNIT' AND [OBJECT\_ID] = @LotId AND [VAR\_NAME] = '$SMIF';

UPDATE [Karat\_TEST].[dbo].E\_CS\_MFDUNIT\_CUSTOM

SET

[MFD\_REF\_7] =LTRIM(RTRIM(@CSTId))

,[MFD\_REF\_6] =LTRIM(RTRIM(@PodId))

WHERE [MFDUNIT\_ID] = @LotId ;

UPDATE [Karat\_TEST].[dbo].E\_CS\_MFDUNIT\_DETAILS

SET

[REF\_1] =LTRIM(RTRIM(@CSTId))

,[REF\_2] =LTRIM(RTRIM(@PodId))

,[REF\_3] =LTRIM(RTRIM(@LotProtocol))

WHERE [MFDUNIT\_ID] = @LotId AND STATE <> 'TERMINATED';

IF @@ERROR>0

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000)

SET @MessageHeader='ОШИБКА АСССОЦИАЦИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = dbo.Karat\_getSTR\_LotCSTPod(@LotId, @LotProtocol, @CSTId, @CSTProtocol, @PodId, @PodProtocol, 'SHORT')

ROLLBACK TRAN

RETURN

END

UPDATE Karat\_labels

SET [LbParent] = NULL

,[LbName] = @CompNAME

,[LbDate] = getDate()

WHERE RTRIM([LbParent]) IN (RTRIM(@CSTId), RTRIM(@PodId), RTRIM(@lbCodeLOT));

UPDATE Karat\_labels

SET [LbParent] = NULL

,[LbName] = @CompNAME

,[LbDate] = getDate()

WHERE RTRIM([lbCode]) = RTRIM(@PodId);

UPDATE Karat\_labels

SET [LbParent] = LTRIM(RTRIM(@PodId))

,[LbName] = @CompNAME

,[LbDate] = getDate()

WHERE RTRIM([lbCode]) = RTRIM(@CSTId);

UPDATE Karat\_labels

SET [LbParent] = LTRIM(RTRIM(@CSTId))

,[LbName] = @CompNAME --SUBSTRING(@LotProtocol, 1, 3) + ' ' + SUBSTRING(@LotProtocol, 5, 4)

,[LbDate] = getDate()

,[LbOption] = LTRIM(RTRIM(@LotProtocol))

WHERE RTRIM([lbCode]) = RTRIM(@lbCodeLOT);

IF @@ERROR>0

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000)

SET @MessageHeader='ОШИБКА АСССОЦИАЦИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = dbo.Karat\_getSTR\_LotCSTPod(@LotId, @LotProtocol, @CSTId, @CSTProtocol, @PodId, @PodProtocol, 'SHORT')

ROLLBACK TRAN

RETURN

END

COMMIT TRAN

SET @MessageTimeout =5;

SET @MessageSound =2;

SET @MessageStop = 1;

SET @IsCheck =1;

SET @MessageBgColor=CONVERT (int, 0x008050);

SET @MessageHeader=N'АСССОЦИАЦИЯ! '+ISNULL(RTRIM(@MessageHeader),'');

END --- @CommandName='A\_LOT-CST-POD' BEGIN

IF @CommandName='D\_LOT-CST-POD' AND @Param\_counter>=3 BEGIN

-- после проверки @CommandName='C\_LOT-CST-POD'

SET @MessageTimeout =5;

SET @LotId =ISNULL(dbo.Karat\_ER\_Alias\_TO\_MFDUNIT\_ID(@lbCodeLOT),''); -- получаем MFDUNIT\_ID

SET @PODId =ISNULL(@lbCodePOD,''); SET @CSTId =ISNULL(@lbCodeCST,'');

SET @LotProtocol =@lbProtocolLOT; SET @CSTProtocol =@lbProtocolCST; SET @PODProtocol =@lbProtocolPOD;

IF @LotId ='' BEGIN

SET @MessageTimeout =30

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageHeader=N'НЕЗАРЕГИСТРИРОВАННАЯ ПАРТИЯ! '+ISNULL(RTRIM(@MessageHeader),'');

RETURN

END

IF @LotId='' OR @PODId='' OR @CSTId =''

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000)

SET @MessageHeader='ОШИБКА ДИСССОЦИАЦИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = dbo.Karat\_getSTR\_LotCSTPod(@LotId, @LotProtocol, @CSTId, @CSTProtocol, @PodId, @PodProtocol, 'SHORT')

RETURN

END

---------------------------------------------

-- Проверка пройдена

-- Диссоциировать партию/кассету/контейнер

BEGIN TRAN

UPDATE [Karat\_TEST].dbo.Karat\_labels

SET [LbParent] = NULL

,[LbName] = @CompNAME

,[LbDate] = getDate()

WHERE [LBCode] IN (@lbCodeLOT, @CSTId, @PODId)

IF @@ERROR<0

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000)

SET @MessageHeader='ОШИБКА ДИСССОЦИАЦИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = dbo.Karat\_getSTR\_LotCSTPod(@LotId, @LotProtocol, @CSTId, @CSTProtocol, @PodId, @PodProtocol, 'SHORT')

ROLLBACK TRAN

RETURN

END

IF (@LotId IS NOT NULL) BEGIN

UPDATE [Karat\_TEST].[dbo].E\_CS\_VARIABLE\_DETAILS

SET

[VAR\_VALUE] = NULL

WHERE [OBJECT\_TYPE] = 'MFDUNIT' AND [OBJECT\_ID] = @LotId AND ([VAR\_NAME] = '$CASSETTE' OR [VAR\_NAME] = '$SMIF');

UPDATE [Karat\_TEST].[dbo].E\_CS\_MFDUNIT\_CUSTOM

SET [MFD\_REF\_6] = NULL

, [MFD\_REF\_7] = NULL

WHERE LTRIM(RTRIM(MFDUNIT\_ID)) = LTRIM(RTRIM(@LOTID));

UPDATE [Karat\_TEST].[dbo].E\_CS\_MFDUNIT\_DETAILS

SET [REF\_1] = NULL

, [REF\_2] = NULL

, [REF\_3] = NULL

WHERE LTRIM(RTRIM(MFDUNIT\_ID)) = LTRIM(RTRIM(@LOTID)) --AND STATE <> 'TERMINATED';

END;

IF @@ERROR<0

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader='ОШИБКА ДИСССОЦИАЦИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = dbo.Karat\_getSTR\_LotCSTPod(@LotId, @LotProtocol, @CSTId, @CSTProtocol, @PodId, @PodProtocol, 'SHORT')

ROLLBACK TRAN

RETURN

END

COMMIT TRAN

SET @MessageTimeout =5;

SET @MessageSound =2;

SET @MessageStop = 1;

SET @IsCheck =1;

SET @MessageBgColor=CONVERT (int, 0x008050);

SET @MessageHeader=N'ДИССОЦИАЦИЯ! '+ISNULL(RTRIM(@MessageHeader),'');

END -- @CommandName='D\_LOT-CST-POD' BEGIN

IF @CommandName='I\_LOT-CST-POD' BEGIN

-- Ввод в таблицу Karat\_Labels новых объектов 'LOT-CST-POD'

SET @MessageTimeout =5;

IF @Param1 NOT IN ('LOT\_FEOL','LOT\_MOL','LOT\_BEOL','CST\_FEOL','CST\_MOL','CST\_BEOL','POD\_FEOL','POD\_MOL','POD\_BEOL')

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader=N'ОШИБКА ПРОТОКОЛА LBCODE! '+ ISNULL(@MessageHeader, '');

SET @MessageText = N'ПРОТОКОЛ '+ISNULL(@Param1,'')

RETURN

END

IF SUBSTRING(@Param1, 1, 3)='LOT' SET @LbMaterial=N'ПАРТИЯ'

ELSE SET @LbMaterial=N'КОД'

IF @Param\_counter>1 BEGIN

IF @LbMaterial=N'ПАРТИЯ' AND (SELECT COUNT(\*) FROM KARAT\_TEST.dbo.E\_CS\_MFDUNIT\_DETAILS WHERE MFDUNIT\_ID = @ParamCurrent)<1

IF (SELECT COUNT(\*) FROM KARAT\_TEST.dbo.Karat\_Labels WHERE LbCode IN (@LbCode))>0 BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader=N'ОШИБКА НЕСУЩЕСТВУЮЩЕЙ ПАРТИИ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = ISNULL(@Param1,'')+' '+ISNULL(@LbCode,'')

RETURN

END

SET @LbType=SUBSTRING(@Param1, 1, 3);

SET @LbCode=@ParamCurrent

--- Проверка, что такой LbCode не существует

IF (SELECT COUNT(\*) FROM KARAT\_TEST.dbo.Karat\_Labels WHERE LbCode IN (@LbCode))>0 BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader=N'ОШИБКА ДУБЛИКАТ LBCODE! '+ ISNULL(@MessageHeader, '');

SET @MessageText = ISNULL(@Param1,'')+' '+ISNULL(@LbCode,'')

RETURN

END

BEGIN TRAN

INSERT INTO KARAT\_TEST.dbo.Karat\_labels (

[lbCode\_CHAR4]

,[lbCode]

,[lbMaterial]

,[lbProducer]

,[lbType]

,[lbOption]

,[lbModLocCode]

,[lbGroup]

,[lbDestinationCode]

,[lbFixedChar]

,[lbWastedChar]

,[lbName]

,[lbParent]

,[lbFact\_Place]

,[lbDate]

,[lbStatus]

)

VALUES (N'0000', @LbCode, @LbMaterial, N'NMT', @LbType, @Param1, 0, 0, NULL, LEN(@LbCode), 0, @CompName, NULL, NULL, getDate(), N'ACTIVE');

IF @@ERROR<0

BEGIN

SET @MessageTimeout =20;

SET @MessageSound =1;

SET @MessageStop = 1;

SET @MessageBgColor=CONVERT (int, 0x770000);

SET @MessageHeader= N'ОШИБКА ДАННЫХ! '+ ISNULL(@MessageHeader, '');

SET @MessageText = ISNULL(@LbCode,'') + ' '+ ISNULL(@LBType,'') + ISNULL(@MessageText, '');

ROLLBACK TRAN

RETURN

END

COMMIT TRAN

END

END --@CommandName='I\_LOT-CST-POD' BEGIN

END

-- ============================================

--EXECUTE mindeo\_command @CommandString = N'Test!';

USE [KARAT\_TEST]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[mindeo\_Input] Script Date: 11.12.2024 19:35:36 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[mindeo\_Input]

(

@CommandString nvarchar(1024) = NULL

)

AS

BEGIN

DECLARE @CommandStringCST nvarchar(1024) = '192.168.1.101;DT-AMORENKO;AW0000T0000$;CST\_FEOL;F1X;';

DECLARE @x int = 0, @y int = 0;

SET @x=0;

WHILE @x<2 BEGIN

SET @CommandString= REPLACE(@CommandStringCST,'X',CAST(@x AS NCHAR(1)));

EXEC [dbo].[mindeo\_command] @CommandString =@CommandString;

SET @x = @x+1;

END

END