

Report

Schleder
04/04/2006

Conteúdo

1 Pacote SysPed.....	3
1.1 Diagrama SysPed.....	3
1.1.1 Card of diagram SysPed.....	10
1.2 List of tables.....	10
1.3 Tabelas.....	12
1.3.1 Tabela DESCONTOS.....	12
1.3.2 Tabela FINALIZADORAS.....	14
1.3.3 Tabela GRUPOS_MOVIMENTOS.....	17
1.3.4 Tabela PARAMETROS_PED.....	21
1.3.5 Tabela PEDIDOS.....	23
1.3.6 Tabela PEDIDOS_DESCONTOS.....	29
1.3.7 Tabela PEDIDOS_ENTREGA.....	31
1.3.8 Tabela PEDIDOS_HISTORICOS.....	33
1.3.9 Tabela PEDIDOS_ITENS.....	35
1.3.10 Tabela PEDIDOS_ITENS_CONF.....	39
1.3.11 Tabela PEDIDOS_MENSAGENS.....	42
1.3.12 Tabela PEDIDOS_PARCELAS.....	44
1.3.13 Tabela PEDIDOS_PENDENCIAS.....	46
1.3.14 Tabela PEDIDOS_VINCULOS.....	48
1.3.15 Tabela TIPO_DESCONTOS.....	50
1.3.16 Tabela TIPO_FRETES.....	53
1.3.17 Tabela TIPO_MOVIMENTOS.....	55
1.3.18 Tabela TIPO_PAGAMENTOS.....	58
1.3.19 Tabela TIPO_PEDIDOS.....	62
1.3.20 Tabela TIPO_PRAZO_ENTREGA.....	64
1.3.21 Tabela TIPO_STATUS_PEDIDOS.....	66
1.3.22 Tabela VINCULO_TPGTO_FINALIZ.....	69
1.4 Stored Procedures.....	71
1.4.1 Procedure STP_GET_CONF_COMPO_TREF.....	71
1.4.2 Procedure STP_GET_PK_DESCONTOS.....	74
1.4.3 Procedure STP_GET_PK_ORDER.....	75
1.4.4 Procedure STP_GET_PK_PEDIDOS_DESCONTOS.....	77
1.4.5 Procedure STP_GET_PK_PEDIDOS_HISTORICOS.....	79
1.4.6 Procedure STP_GET_PK_PRODUCT.....	81
1.4.7 Procedure STP_GET_PRODUCT_ATAxes.....	82
1.4.8 Procedure STP_GET_PRODUCT_BOX.....	88
1.4.9 Procedure STP_GET_PRODUCT_ITEM.....	90
1.4.10 Procedure STP_PROMOVE_PEDIDO.....	103
1.4.11 Procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS.....	105

1 Pacote SysPed

1.1 Diagrama SysPed

Módulo de Gerenciamento de Pedidos

STP_GET_PRODUCT_ATAxes

STP_GET_PK_PEDIDOS_DESCONTOS

STP_GET_PK_DESCONTOS

STP_GET_PK_PEDIDOS_HISTORICOS

STP_GET_PRODUCT_ITEM

STP_GET_PK_PRODUCT

STP_PROMOVE_PEDIDO

STP_GET_CONF_COMPO_TREF

STP_GET_PK_ORDER

DESCONTOS		
FK_TIPO_DESCONTOS	VALORS	<pk,fk1>
PK_DESCONTOS	VALORS	<pk>
FK_CATEGORIAS	VALORS	<fk2>
FK_PAISES	VALORI	<fk3>
FK_ESTADOS	UF	<fk3>
IDX_DSCT	NUMERO_INDICE	
FLAG_TDSC	FLAG_TIPO_DESCONTO	
FLAG_DSTQ	FLAG_SIM_NAO	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

TIPO_DESCONTOS		
PK_TIPO_DESCONTOS	VALORS	<pk>
FK_GRUPOS_MOVIMENTOS	VALORS	<fk>
DSC_TDSC	DESCRICAO_30RQ	
KC_DESCONTOS	VALORS	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

STP_UPDATE_SUPPLIERS_FROM_ITEMS

STP_GET_PRODUCT_BOX

TIPO_FRETES		
PK_TIPO_FRETES	VALORS	<pk>
FK_TIPO_PAGAMENTOS	VALORS	<fk>
FK_CLASSIFICACAO	VALORS	
DSC_TPFRE	DESCRICAO_30RQ	
FLAG_TP_FRE	FLAG_TIPO_FRETE	
FLAG_FIN	FLAG_SIM_NAO	
FLAG_ACU	FLAG_SIM_NAO	
FLAG_NF	FLAG_SIM_NAO	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

TIPO_PAGAMENTOS		
PK_TIPO_PAGAMENTOS	VALORS	<pk>
FK_GRUPOS_MOVIMENTOS	VALORS	<fk2>
FK_TIPO_DESCONTOS	VALORS	<fk1>
DSC_TPG	DESCRICAO_50RQ	
LST_PRZ	DESCRICAO_50	
FLAG_SEN	FLAG_SIM_NAO	
FLAG_TINT	FLAG_PARCELA	
FLAG_TVDA	FLAG_TIPO_VENDA	
FLAG_ALT_LST	FLAG_SIM_NAO	
FLAG_DATA_BASE	FLAG_DATA_BASE	
FLAG_DEF_USER	FLAG_SIM_NAO	
QTD_PAR	VALORS	
INTRV_TAB	VALORS	
IND_TPGT	NUMERO_INDICE	
PER_JUR	NUMERO_PERCENTUAL	
DSCT_TPG	NUMERO_INDICE	
IND_ENTR	NUMERO_INDICE	
QTD_ENTR	VALORS	
PER_ENTR	NUMERO_PERCENTUAL	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

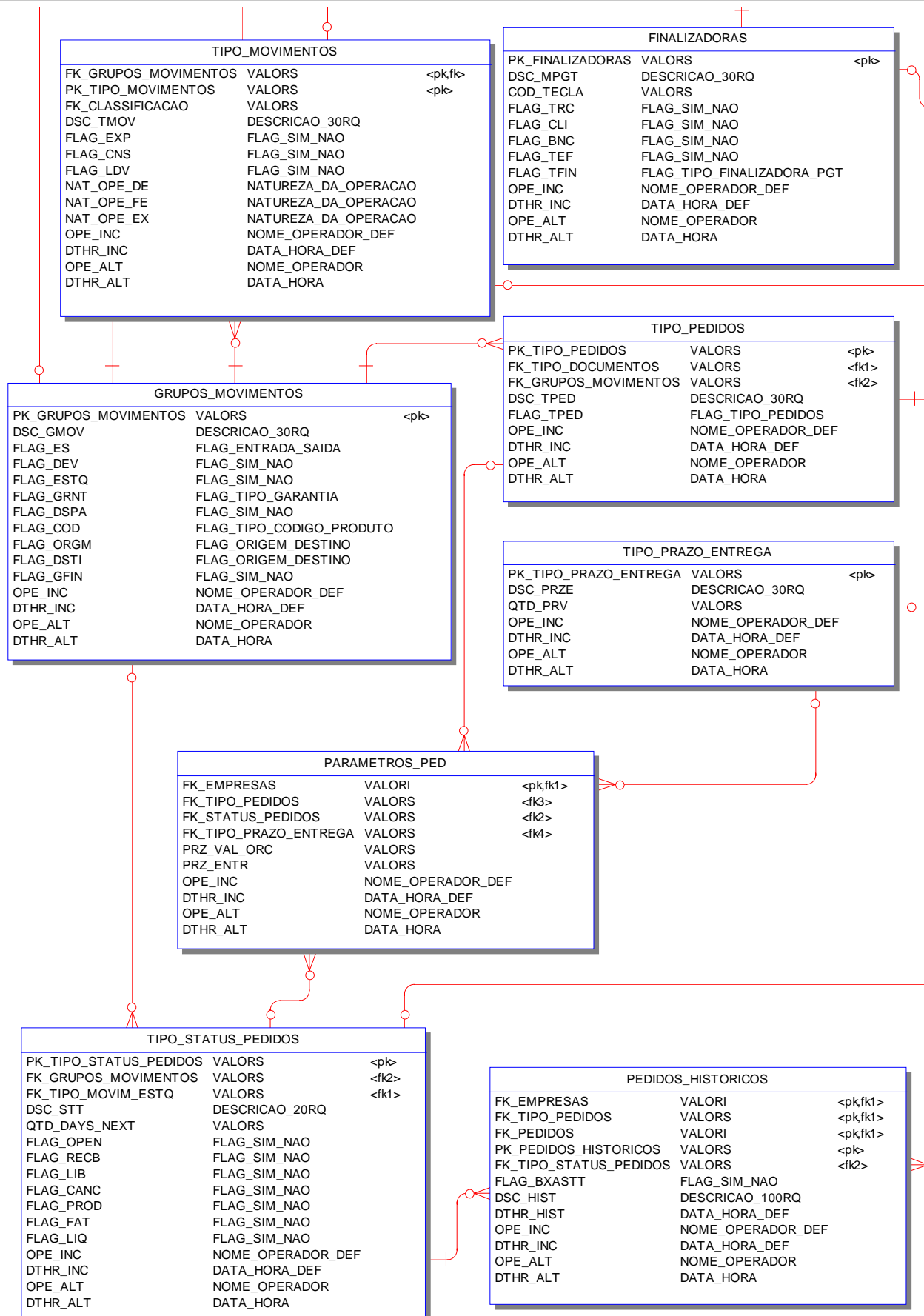
VINCULO_TPGTO_FINALIZ		
FK_FINALIZADORAS	VALORS	<pk,fk1>
FK_TIPO_PAGAMENTOS	VALORS	<pk,fk2>
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

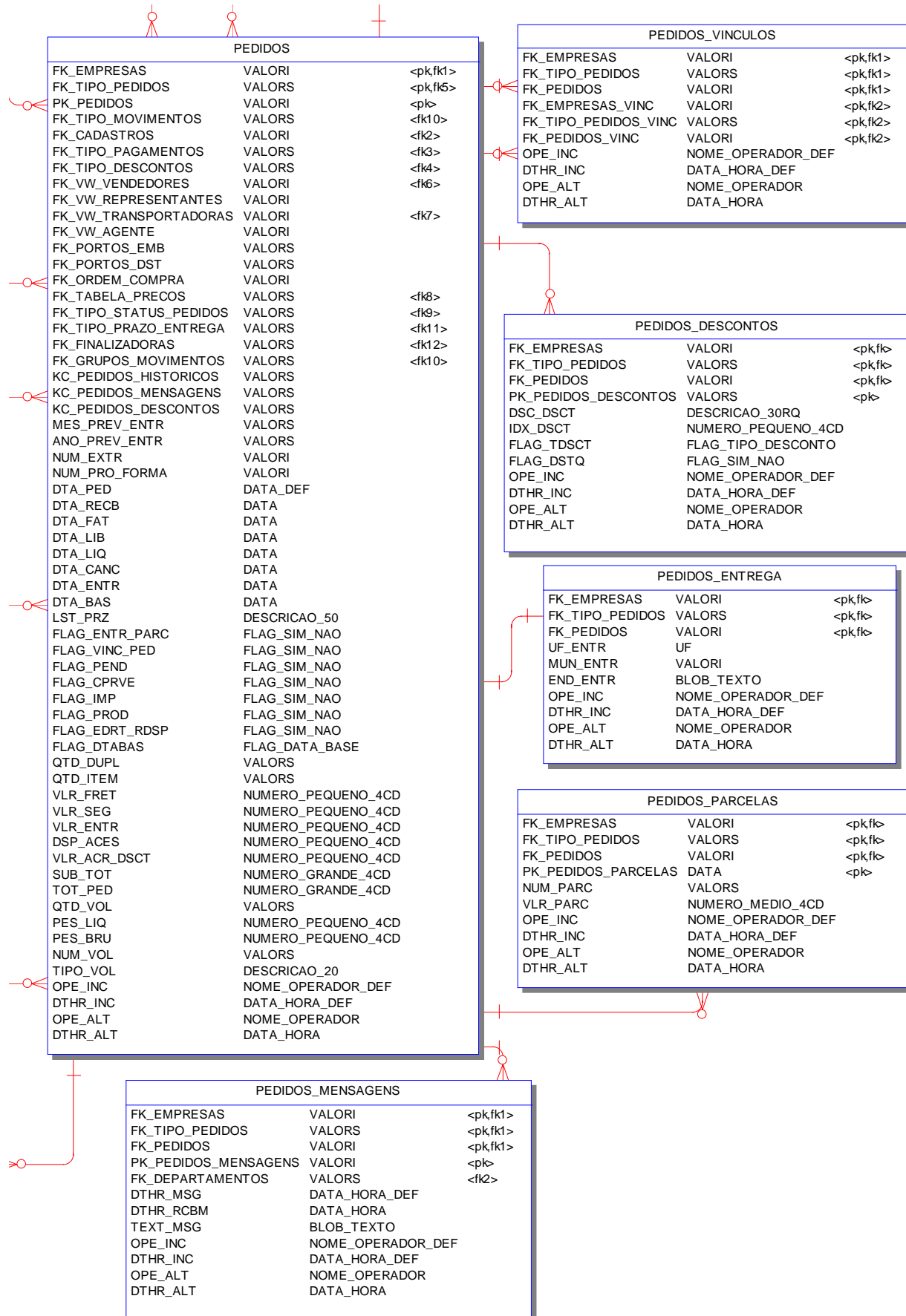
PEDIDOS_PENDENCIAS		
FK_EMPRESAS	VALORI	<pk,fk>
FK_TIPO_PEDIDOS	VALORS	<pk,fk>
FK_PEDIDOS	VALORI	<pk,fk>
FK_PEDIDOS_ITENS	VALORS	<pk,fk>
FK_PRODUTOS	VALORI	<pk>
QTD_PEND	NUMERO_PEQUENO_4CD	
VLR_UNIT	NUMERO_PEQUENO_4CD	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

PEDIDOS_ITENS_CONF		
FK_EMPRESAS	VALORI	<pk,fk1,fk3>
FK_TIPO_PEDIDOS	VALORS	<pk,fk1>
FK_PEDIDOS	VALORI	<pk,fk1>
FK_PEDIDOS_ITENS	VALORS	<pk,fk1>
PK_PEDIDOS_ITENS_CONF	VALORS	<pk>
FK_PRODUTOS	VALORI	<fk2,fk3>
FK_COMPONENTES	VALORS	<fk3>
FK_ACABAMENTOS	VALORS	<fk3>
FK_TIPO_REFERENCIAS	VALORS	<fk3>
FK_INSUMOS	VALORI	
COD_REF	CODIGO_PRODUTO	
VLR_ITM	NUMERO_MEDIO_4CD	
QTD_COMP	VALORS	
QTD_COMP_TOT	VALORS	
QTD_INS	NUMERO_PEQUENO_4CD	
FLAG_FRNCLI	FLAG_SIM_NAO	
FLAG_PEND	FLAG_SIM_NAO	
FLAG_CNTR	FLAG_SIM_NAO	
FLAG_BXASTT	FLAG_SIM_NAO	
FK_TIPO_DOCUMENTOS	VALORS	
NUM_DOC_LIB	VALORI	
PER_DSCT_INS	NUMERO_PERCENTUAL	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_INC	DATA_HORA_DEF	
OPE_ALT	NOME_OPERADOR	
DTHR_ALT	DATA_HORA	

Physical Data Model	
Model:	PROCESSA_FB
Package:	SysPed
Diagram:	SysPed
Author:	Alcindo Schleder
Date:	03/04/2006
Version:	1.0.2.0

PEDIDOS_ITENS		
FK_EMPRESAS	VALORI	<pk,fk1,fk5,fk8>
FK_TIPO_PEDIDOS	VALORS	<pk,fk1>
FK_PEDIDOS	VALORI	<pk,fk1>
PK_PEDIDOS_ITENS	VALORS	<pk>
FK_TABELA_PRECOS	VALORS	<fk4>
FK_GRUPOS_MOVIMENTOS	VALORS	<fk3>
FK_TIPO_MOVIMENTOS	VALORS	<fk3>
FK_UNIDADES	VALORS	<fk2>
FK_CARGAS_PRODUCAO	VALORI	<fk5>
FK_PRODUTOS	VALORI	<fk6>
FK_CLASSIFICACAO	VALORI	<fk7>
FK_ALMOXARIFADOS	VALORS	<fk8>
FK_LOTACOES	VALORS	<fk8>
REF_PRODUTO	CODIGO_PRODUTO	
LOT_ITEM	DESCRICAO_10	
QTD_ITEM	NUMERO_PEQUENO_4CD	
VLR_TAB	NUMERO_PEQUENO_4CD	
VLR_UNIT	NUMERO_PEQUENO_4CD	
SUB_TOT	NUMERO_MEDIO_4CD	
VLR_ACR_DSCT	NUMERO_PEQUENO_4CD	
TOT_ITEM	NUMERO_MEDIO_4CD	
SIT_TRIB	SITUACAO_TRIBUTARIA	
ALQ_ISS	NUMERO_PERCENTUAL	
ALQ_ICMS	NUMERO_PERCENTUAL	
ALQ_ICMSS	NUMERO_PERCENTUAL	
ALQ_IPI	NUMERO_PERCENTUAL	
ALQ_OTR	NUMERO_PERCENTUAL	
COD_ISS_ECF	CODIGO_IMPOSTO_ECF	
COD_ICMS_ECF	CODIGO_IMPOSTO_ECF	
COD_ICMSS_ECF	CODIGO_IMPOSTO_ECF	
COD_IPI_ECF	CODIGO_IMPOSTO_ECF	
COD_OTR_ECF	VALORS	
FLAG_PRM	FLAG_SIM_NAO	
FLAG_PEND	FLAG_SIM_NAO	
FLAG_BXASTT	FLAG_SIM_NAO	
FLAG_SRV	FLAG_SIM_NAO	
FLAG_CONF	FLAG_SIM_NAO	
DTA_FAT	DATA	
DTA_LIQ	DATA	
NUM_EXT	VALORI	
OPE_INC	NOME_OPERADOR_DEF	
DTHR_ALT	DATA_HORA	
OPE_ALT	NOME_OPERADOR	
DTHR_INC	DATA_HORA_DEF	





1.1.1 Card of diagram SysPed

<i>Name</i>	SysPed
<i>Code</i>	SYSPED
<i>Comment</i>	

1.2 List of tables

Name	Comment
PEDIDOS_PENDENCIAS	Tabela que armazena uma estatística de produtos que foram pedidos e não encontrados em estoques será gerada pela procedure STP_BUSCA_PRODUTOS e STP_BUSCA_CONFIGURACOES
PEDIDOS	Arquivo que Armazena todos os dados do cabeçalho do pedido. Este arquivo poderá armazenar pedidos de Entradas e pedidos de Saídas, sendo estes diferenciados pelo Tipo de Movimento que será atribuído a cada Documento. Nos Pedidos de Entradas ele armazenará o o número do pedido conforme o número dado pelo fornecedor para posterior identificação e baixa na entrada das mercadorias.... Deve-se fazer uma tela como se fosse uma Nota Fiscal.... que calcula todos os campos e tudo em memória para depois descarregá-lo no disco gerando o número do Pedido (No caso de pedido de entrada).
PEDIDOS_ITENS	Tabela que armazena os itens dos pedidos....
PEDIDOS_ITENS_CONF	Tabela que armazena as configurações dos Ítems do Pedido. Tecido; Cor; etc...
GRUPOS_MOVIMENTOS	Tabela que gerencia como se comportarão os vários documentos de entradas e saídas de mercadorias identificando que tipo de movimentação ela fará nestes documentos.
TIPO_DESCONTOS	Tabela que informa o tipo de desconto do fornecedor
DESCONTOS	Tabela que armazena os índices de descontos
TIPO_FRETES	Tabela que Armazena os tipos de Fretes do Sistema
TIPO_PAGAMENTOS	Tabela que armazena os prazos defaults para vendas e/ou compras

Name	Comment
TIPO_MOVIMENTOS	Tabela que identifica qual vai ser a Natureza de operação dos documentos fiscais....
TIPO_PEDIDOS	Tabela que armazena os tipos de pedidos que podem ser feitos
TIPO_STATUS_PEDIDOS	Tabela que define o status do pedido
TIPO_PRAZO_ENTREGA	Tabela que informa os tipos de prazos de entrega do pedido
FINALIZADORAS	Tabela que informa o meio de pagamento do pedidos
VINCULO_TPGTO_FINALIZ	Vincula os tipos de pagamento às finalizadoras
PEDIDOS_VINCULOS	
PEDIDOS_ENTREGA	tabela que armazena os endereços de entrega e/ou redespacho
PEDIDOS_HISTORICOS	Tabela que armazena o histórico dos status do pedido. Esta tabela é usada para controlar se algum status já foi usado na tabela fazendo que a banxa de um status anterior não seja feita duas vezes...
PEDIDOS_MENSAGENS	
PEDIDOS_DESCONTOS	Tabela que armazena todos os descontos dados pelo sistema e usuário
PARAMETROS_PED	Tabela que armazena os parâmentos dos pedidos
PEDIDOS_PARCELAS	Tabela que armazena as parcelas geradas para o faturamento previsto do pedido

1.3 Tabelas

1.3.1 Tabela DESCONTOS

1.3.1.1 Card of table DESCONTOS

Name	DESCONTOS
Code	DESCONTOS
Comment	Tabela que armazena os índices de descontos
Primary Key	PK_DESCONTOS

1.3.1.2 Lista de Colunas da Tabela DESCONTOS

Name	Data Type	Domain	Comment
FK_TIPO_D ESCONTOS	SMALLINT	VALORS	Código do tipo do desconto
PK_DESCO NTOS	SMALLINT	VALORS	Código do desconto
FK_CATEGO RIAS	SMALLINT	VALORS	Categoria do cliente que deve receber este desconto;
FK_PAISES	INTEGER	VALORI	Código do País
FK_ESTADO S	CHAR(2)	UF	Símbolo da unidade federativa do país
IDX_DSCT	NUMERIC(9,4)	NUMERO_INDICE	
FLAG_TDSC T	SMALLINT	FLAG_TIPO_DESCONTO	Flag que indica o tipo do desconto aplicado à uma operação 0 ==> Percentual 1 ==> Índice Multiplicador 2 ==> Índice Divisor 3 ==> Valor
FLAG_DSTQ	SMALLINT	FLAG_SIM_NAO	Flag que indica se o desconto irá ser destacado no documento ou soma com os outros
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.1.3 Dependências da Tabela DESCONTOS

Name	Code	Class Name
FK_DESCONTOS_TIPO_DESCON TO	FK_DESCONTOS_TIPO_DESCON TO	Reference
FK_DESCONTOS_CATEGORIAS	FK_DESCONTOS_CATEGORIAS	Reference
FK_DESCONTOS_ESTADOS	FK_DESCONTOS_ESTADOS	Reference

Name	Code	Class Name
DESCONTOS	DESCONTOS	Shortcut of table

1.3.1.4 Lista de Chaves da Tabela DESCONTOS

Name	Code	Primary
PK_DESCONTOS	PK_DESCONTOS	X

1.3.1.5 Chave PK_DESCONTOS da Tabela DESCONTOS

1.3.1.5.1 Lista de Colunas da Chave PK_DESCONTOS

Name	Code
FK_TIPO_DESCONTOS	FK_TIPO_DESCONTOS
PK_DESCONTOS	PK_DESCONTOS

1.3.1.5.2 Dependências da Chave PK_DESCONTOS

Name	Code	Class Name
PK_DESCONTOS	PK_DESCONTOS	Index

1.3.1.6 Lista de Índices DESCONTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_DESCONTOS	PK_DESCONTOS	X		X		

1.3.1.7 Índice PK_DESCONTOS da Tabela DESCONTOS

1.3.1.7.1 Lista de Colunas do Índice PK_DESCONTOS

Name	Code	Sort
FK_TIPO_DESCONTOS	FK_TIPO_DESCONTOS	Ascending
PK_DESCONTOS	PK_DESCONTOS	Ascending

1.3.1.8 Lista de Triggers da Tabela DESCONTOS

Name	Code
TBIG_DESCONTOS	TBIG_DESCONTOS
TBU0_DESCONTOS	TBU0_DESCONTOS

1.3.2 Tabela FINALIZADORAS

1.3.2.1 Card of table FINALIZADORAS

Name	FINALIZADORAS
Code	FINALIZADORAS
Comment	Tabela que informa o meio de pagamento do pedidos
Primary Key	PK_FINALIZADORAS

1.3.2.2 Lista de Colunas da Tabela FINALIZADORAS

Name	Data Type	Domain	Comment
PK_FINALIZADORAS	SMALLINT	VALORS	Código
DSC_MPGT	VARCHAR(30)	DESCRICAO_30RQ	Descrição
COD_TECLA	SMALLINT	VALORS	Código da telcla a ser utilizada no sistema
FLAG_TRC	SMALLINT	FLAG_SIM_NAO	Flag que indica se este tipo de finalizadora pode devolver troco
FLAG_CLI	SMALLINT	FLAG_SIM_NAO	Flag que indica se a finalizadora exige o código do emitente
FLAG_BNC	SMALLINT	FLAG_SIM_NAO	Flag que indica que se deve informar um banco na finalizadora
FLAG_TEF	SMALLINT	FLAG_SIM_NAO	Flag que indica se será feita transferência eletrônica de fundos (TEF) quando feita a finalizadora que requer um resultado para fechar o faturamento
FLAG_TFIN	SMALLINT	FLAG_TIPO_FINALIZADORA_PGT	Flag que indica o Tipo da Finalizadora 0 ==> Dinheiro 1 ==> Cheque 2 ==> Cheque Pré-Datado 3 ==> Cartão de Crédito 4 ==> Cartão de Débito 5 ==> Depósito Bancário / DOC 6 ==> Boleto Bancário 7 ==> Cobrança em Carteira 8 ==> Nenhum
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.2.3 Dependências da Tabela FINALIZADORAS

Name	Code	Class Name
FK_VINCULO_TPGTO_FINALIZADORAS	FK_VINCULO_TPGTO_FINALIZADORAS	Reference

Name	Code	Class Name
FK_PEDIDOS_FINALIZADORAS	FK_PEDIDOS_FINALIZADORAS	Reference
FK_PRACAS_OCORRE_FINALIZADORAS	FK_PRACAS_OCORRE_FINALIZADORAS	Reference
FK_CONTAS_LANCAM_FINALIZADORAS	FK_CONTAS_LANCAM_FINALIZADORAS	Reference
FK_FINALIZADORAS_CLIENTES	FK_FINALIZADORAS_CLIENTES	Reference
FINALIZADORAS	FINALIZADORAS	Shortcut of table
FINALIZADORAS	FINALIZADORAS	Shortcut of table
FINALIZADORAS	FINALIZADORAS	Shortcut of table

1.3.2.4 Lista de Chaves da Tabela FINALIZADORAS

Name	Code	Primary
PK_FINALIZADORAS	PK_FINALIZADORAS	X

1.3.2.5 Chave PK_FINALIZADORAS da Tabela FINALIZADORAS

1.3.2.5.1 Lista de Colunas da Chave PK_FINALIZADORAS

Name	Code
PK_FINALIZADORAS	PK_FINALIZADORAS

1.3.2.5.2 Dependências da Chave PK_FINALIZADORAS

Name	Code	Class Name
PK_FINALIZADORAS	PK_FINALIZADORAS	Index
FK_FINALIZADORAS_CLIENTES	FK_FINALIZADORAS_CLIENTES	Reference
FK_VINCULO_TPGTO_FINALIZADORAS	FK_VINCULO_TPGTO_FINALIZADORAS	Reference
FK_PEDIDOS_FINALIZADORAS	FK_PEDIDOS_FINALIZADORAS	Reference
FK_CONTAS_LANCAM_FINALIZADORAS	FK_CONTAS_LANCAM_FINALIZADORAS	Reference
FK_PRACAS_OCORRE_FINALIZADORAS	FK_PRACAS_OCORRE_FINALIZADORAS	Reference

1.3.2.6 Lista de Índices FINALIZADORAS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_FINALIZADORAS	PK_FINALIZADORAS	X		X		

1.3.2.7 Índice PK_FINALIZADORAS da Tabela FINALIZADORAS

1.3.2.7.1 Lista de Colunas do Índice PK_FINALIZADORAS

Name	Code	Sort
PK_FINALIZADORAS	PK_FINALIZADORAS	Ascending

1.3.2.8 Lista de Triggers da Tabela FINALIZADORAS

Name	Code
TBIG_FINALIZADORAS	TBIG_FINALIZADORAS
TBU0_FINALIZADORAS	TBU0_FINALIZADORAS

1.3.3 Tabela GRUPOS_MOVIMENTOS

1.3.3.1 Card of table GRUPOS MOVIMENTOS

Name	GRUPOS_MOVIMENTOS
Code	GRUPOS_MOVIMENTOS
Comment	Tabela que gerencia como se comportarão os vários documentos de entradas e saídas de mercadorias identificando que tipo de movimentação ela fará nestes documentos.
Primary Key	PK_GRUPOS_MOVIMENTOS

1.3.3.2 Lista de Colunas da Tabela GRUPOS_MOVIMENTOS

Name	Data Type	Domain	Comment
PK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
DSC_GMOV	VARCHAR(30)	DESCRICAO_30RQ	Descrição do Tipo de Movimento
FLAG_ES	SMALLINT	FLAG_ENTRADA_SAIDA	Flag que identifica o Movimento como sendo de Entrada ou Saída de Mercadorias (Compras ou Vendas). 0 ==> Entradas 1 ==> Saídas
FLAG_DEV	SMALLINT	FLAG_SIM_NAO	Flag que indica que o documento é um documento de devolução de Mercadorias
FLAG_ESTQ	SMALLINT	FLAG_SIM_NAO	Flag que indica se o movimento atualiza estoques
FLAG_GRNT	SMALLINT	FLAG_TIPO_GARANTIA	Flag que indica que o movimento é uma devolução para garantia, como será cobrada (se for), e de quem. 0 ==> Nenhuma 1 ==> por conta do Fornecedor 2 ==> por conta do Transportador 3 ==> por conta do Cliente
FLAG_DSPA	SMALLINT	FLAG_SIM_NAO	Flag que indica se o frete será calculado como despesas acessórias
FLAG_COD	SMALLINT	FLAG_TIPO_CODIGO_PRODUTO	Flag que Indica qual o tipo de Código será usado para o movimento: 0 ==> Referência do Produto 1 ==> Código do Produto 2 ==> Código de Barras 3 ==> Código do Fornecedor

Name	Data Type	Domain	Comment
FLAG_ORGM	SMALLINT	FLAG_ORIGEM_DESTINO	Flag da Origem do Documento 0 ==> Estoques 1 ==> Clientes 2 ==> Fornecedores 3 ==> Controle Interno 4 ==> Filiais
FLAG_DSTI	SMALLINT	FLAG_ORIGEM_DESTINO	Flag que indica qual o destino dos produtos discriminados no documento: 0 ==> Estoques 1 ==> Clientes 2 ==> Fornecedores 3 ==> Controle Interno 4 ==> Filiais
FLAG_GFIN	SMALLINT	FLAG_SIM_NAO	Flag que indica se o movimento irá gerar movimentação financeira
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.3.3 Dependências da Tabela GRUPOS_MOVIMENTOS

Name	Code	Class Name
FK_TIPO_MOVIMENT_GRUPOS_MOVIME	FK_TIPO_MOVIMENT_GRUPOS_MOVIME	Reference
FK_TIPO_PEDIDOS_GRUPOS_MOVIME	FK_TIPO_PEDIDOS_GRUPOS_MOVIME	Reference
FK_TIPO_PAGAMENT_GRUPOS_MOVIME	FK_TIPO_PAGAMENT_GRUPOS_MOVIME	Reference
FK_TIPO_DESCONTO_GRUPOS_MOVIME	FK_TIPO_DESCONTO_GRUPOS_MOVIME	Reference
FK_TIPO_STATUS_GRUPOS_MOVIM	FK_TIPO_STATUS_GRUPOS_MOVIM	Reference
FK_TIPO_ORDENS_S_GRUPOS_MOVIME	FK_TIPO_ORDENS_S_GRUPOS_MOVIME	Reference
FK_DOCUMENTOS_GRUPOS_MOVIME	FK_DOCUMENTOS_GRUPOS_MOVIME	Reference
FK_CANCELAMENTOS_TIPO_MOVIMENT	FK_CANCELAMENTOS_TIPO_MOVIMENT	Reference
GRUPOS_MOVIMENTOS	GRUPOS_MOVIMENTOS	Shortcut of table

Name	Code	Class Name
GRUPOS_MOVIMENTOS	GRUPOS_MOVIMENTOS	Shortcut of table

1.3.3.4 Lista de Chaves da Tabela GRUPOS_MOVIMENTOS

Name	Code	Primary
PK_GRUPOS_MOVIMENTOS	PK_GRUPOS_MOVIMENTOS	X

1.3.3.5 Chave PK_GRUPOS_MOVIMENTOS da Tabela GRUPOS_MOVIMENTOS

1.3.3.5.1 Lista de Colunas da Chave PK_GRUPOS_MOVIMENTOS

Name	Code
PK_GRUPOS_MOVIMENTOS	PK_GRUPOS_MOVIMENTOS

1.3.3.5.2 Dependências da Chave PK_GRUPOS_MOVIMENTOS

Name	Code	Class Name
PK_GRUPOS_MOVIMENTOS	PK_GRUPOS_MOVIMENTOS	Index
FK_TIPO_MOVIMENT_GRUPOS_MO OVIME	FK_TIPO_MOVIMENT_GRUPOS_M OVIME	Reference
FK_TIPO_PEDIDOS_GRUPOS_MO VIME	FK_TIPO_PEDIDOS_GRUPOS_MO VIME	Reference
FK_TIPO_PAGAMENT_GRUPOS_ MOVIME	FK_TIPO_PAGAMENT_GRUPOS_ MOVIME	Reference
FK_TIPO_DESCONTO_GRUPOS_ MOVIME	FK_TIPO_DESCONTO_GRUPOS_ MOVIME	Reference
FK_TIPO_STATUS_GRUPOS_MOV IM	FK_TIPO_STATUS_GRUPOS_MOV IM	Reference
FK_TIPO_ORDENS_S_GRUPOS_ MOVIME	FK_TIPO_ORDENS_S_GRUPOS_ MOVIME	Reference
FK_DOCUMENTOS_GRUPOS_MO VIME	FK_DOCUMENTOS_GRUPOS_MO VIME	Reference
FK_CANCELAMENTOS_TIPO_MO VIMENT	FK_CANCELAMENTOS_TIPO_MO VIMENT	Reference

1.3.3.6 Lista de Índices GRUPOS_MOVIMENTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_GRUPOS_MOVIMENTOS	PK_GRUPOS_MOVIMENTOS	X		X		

1.3.3.7 Índice PK_GRUPOS_MOVIMENTOS da Tabela GRUPOS_MOVIMENTOS

1.3.3.7.1 Lista de Colunas do Índice PK_GRUPOS_MOVIMENTOS

Name	Code	Sort
------	------	------

Name	Code	Sort
PK_GRUPOS_MOVIMENTOS	PK_GRUPOS_MOVIMENTOS	Ascending

1.3.3.8 Lista de Triggers da Tabela GRUPOS_MOVIMENTOS

Name	Code
TBIG_GRUPOS_MOVIMENTOS	TBIG_GRUPOS_MOVIMENTOS
TBU0_GRUPOS_MOVIMENTOS	TBU0_GRUPOS_MOVIMENTOS

1.3.4 Tabela PARAMETROS_PED

1.3.4.1 Card of table PARAMETROS_PED

Name	PARAMETROS_PED
Code	PARAMETROS_PED
Comment	Tabela que armazena os parâmetros dos pedidos
Primary Key	PK_PARAMETROS_PED

1.3.4.2 Lista de Colunas da Tabela PARAMETROS_PED

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Tipo de Pedidos Default
FK_STATUS_PEDIDOS	SMALLINT	VALORS	Status do pedido default
FK_TIPO_PRAZO_ENTREGA	SMALLINT	VALORS	Código do Prazo de Entrega default
PRZ_VALOR	SMALLINT	VALORS	Prazo de validade de um orçamento
PRZ_ENTR	SMALLINT	VALORS	Prazo de entrega default
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.4.3 Dependências da Tabela PARAMETROS_PED

Name	Code	Class Name
FK_PARAMETROS_PES_EMPRESAS	FK_PARAMETROS_PES_EMPRESAS	Reference
FK_PARAMETROS_PE_TIPO_STATUS_P	FK_PARAMETROS_PE_TIPO_STATUS_P	Reference
FK_PARAMETROS_PE_TIPO_PEDIDOS	FK_PARAMETROS_PE_TIPO_PEDIDOS	Reference
FK_PARAMETROS_PE_TIPO_PRAZO_EN	FK_PARAMETROS_PE_TIPO_PRAZO_EN	Reference

1.3.4.4 Lista de Chaves da Tabela PARAMETROS_PED

Name	Code	Primary
PK_PARAMETROS_PED	PK_PARAMETROS_PED	X

1.3.4.5 Chave PK_PARAMETROS_PED da Tabela PARAMETROS_PED

1.3.4.5.1 Lista de Colunas da Chave PK_PARAMETROS_PED

Name	Code
FK_EMPRESAS	FK_EMPRESAS

1.3.4.5.2 Dependências da Chave PK_PARAMETROS_PED

Name	Code	Class Name
PK_PARAMETROS_PED	PK_PARAMETROS_PED	Index

1.3.4.6 Lista de Índices PARAMETROS_PED

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PARAMETROS_PED	PK_PARAMETROS_PED	X		X		

1.3.4.7 Índice PK_PARAMETROS_PED da Tabela PARAMETROS_PED

1.3.4.7.1 Lista de Colunas do Índice PK_PARAMETROS_PED

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending

1.3.4.8 Lista de Triggers da Tabela PARAMETROS_PED

Name	Code
TBU0_PARAMETROS_PED	TBU0_PARAMETROS_PED

1.3.5 Tabela PEDIDOS

1.3.5.1 Card of table PEDIDOS

<i>Name</i>	PEDIDOS
<i>Code</i>	PEDIDOS
<i>Comment</i>	Arquivo que Armazena todos os dados do cabeçalho do pedido. Este arquivo poderá armazenar pedidos de Entradas e pedidos de Saídas, sendo estes diferenciados pelo Tipo de Movimento que será atribuído a cada Documento. Nos Pedidos de Entradas ele armazenará o o número do pedido conforme o número dado pelo fornecedor para posterior identificação e baixa na entrada das mercadorias.... Deve-se fazer uma tela como se fosse uma Nota Fiscal.... que calcula todos os campos e tudo em memória para depois descarregá-lo no disco gerando o número do Pedido (No caso de pedido de entrada).
<i>Primary Key</i>	PK_PEDIDOS

1.3.5.2 Lista de Colunas da Tabela PEDIDOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
PK_PEDIDOS	INTEGER	VALORI	Número do Pedido
FK_TIPO_MOVIMENTOS	SMALLINT	VALORS	Tipo de movimento do Pedido. Este campo eh somente usado para referenciar o tipo de movimento default usado na seleção do grupo de movimentos. Não tem efeito na impressão da Nota Fiscal...
FK_CADASTROS	INTEGER	VALORI	
FK_TIPO_PAGAMENTOS	SMALLINT	VALORS	Código do Tipo de Pagamento
FK_TIPO_DESCONTOS	SMALLINT	VALORS	Código do tipo do desconto
FK_VW_VENDEDORES	INTEGER	VALORI	Código do Vendedor
FK_VW_REPRESENTANTES	INTEGER	VALORI	Código do Representante
FK_VW_TRANSPORTADORAS	INTEGER	VALORI	Código do Cadastro
FK_VW_AGENTE	INTEGER	VALORI	Código do Agente Alfandegário
FK_PORTOS_EMB	SMALLINT	VALORS	Código do Porto de Embarque do Agente
FK_PORTOS_DST	SMALLINT	VALORS	Código do porto de destino

Name	Data Type	Domain	Comment
FK_ORDEM_COMPRA	INTEGER	VALORI	Número da Ordem de Compra gerada para o Pedido
FK_TABELA_PRECOS	SMALLINT	VALORS	Código da tabela de preços
FK_TIPO_STATUS_PEDIDOS	SMALLINT	VALORS	Código do Status do Pedido
FK_TIPO_PRAZO_ENTREGA	SMALLINT	VALORS	Código do tipo de prazo de entrega
FK_FINALIZADORAS	SMALLINT	VALORS	Código da finalizadora do pedido
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Grupo de Movimentos
KC_PEDIDOS_HISTORICOS	SMALLINT	VALORS	Key control para a tabela PEDIDOS_HISTORICOS
KC_PEDIDOS_MENSAGENS	SMALLINT	VALORS	Key control para a tabela PEDIDOS_MENSAGENS
KC_PEDIDOS_DESCONTOS	SMALLINT	VALORS	Key Control da Tabela PEDIDOS_DESCONTOS
MES_PREV_ENTR	SMALLINT	VALORS	Mês da previsão de entrega
ANO_PREV_ENTR	SMALLINT	VALORS	Ano da previsão da entrega
NUM_EXTR	INTEGER	VALORI	Número externo informado pelo cliente ou pelo representante
NUM_PROFORMA	INTEGER	VALORI	
DTA_PED	DATE	DATA_DEF	Data da emissão do Pedido
DTA_RECB	DATE	DATA	Data do Recebimento do Pedido
DTA_FAT	DATE	DATA	Data de Faturamento do Pedido
DTA_LIB	DATE	DATA	Data de liberação do Pedido
DTA_LIQ	DATE	DATA	Data de Liquidação do Pedido
DTA_CANC	DATE	DATA	Data de Cancelamento
DTA_ENTR	DATE	DATA	Data de Previsão de Entrega
DTA_BAS	DATE	DATA	Data Base para o Faturamento do Pedido
LST_PRZ	VARCHAR(50)	DESCRICAO_50	
FLAG_ENTR_PARC	SMALLINT	FLAG_SIM_NAO	Flag que indica se o pedido não pode ser entregue parcialmente
FLAG_VINC_PED	SMALLINT	FLAG_SIM_NAO	Flag que indica se há pedidos vinculados a este pedido

Name	Data Type	Domain	Comment
FLAG_PEND	SMALLINT	FLAG_SIM_NAO	Flag que indica que o pedido contém pendências a serem resolvidas
FLAG_CPRV E	SMALLINT	FLAG_SIM_NAO	Flag que indica se necessita a confirmação da entrega do pedido
FLAG_IMP	SMALLINT	FLAG_SIM_NAO	Flag que indica se o pedido já fora impresso
FLAG_PROD	SMALLINT	FLAG_SIM_NAO	Flag que indica se o pedido já está em produção
FLAG_EDRT _RDSP	SMALLINT	FLAG_SIM_NAO	Flag que indica se o pedido contém um endereço para entrega direta ou redespacho
FLAG_DTAB AS	SMALLINT	FLAG_DATA_BASE	Flag que indica a data o tipo da data base para cálculo das datas do financeiro 0 ==> A partir da Data de Faturamento 1 ==> A partir da Data do Pedido 2 ==> A partir da Data de Embarque 3 ==> A partir da Data Informada
QTD_DUPL	SMALLINT	VALORS	Quantidade de Duplicatas que Contém o pedido
QTD_ITEM	SMALLINT	VALORS	Quantidade de itens que contém o pedido
VLR_FRET	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor do Frete do Pedido
VLR_SEG	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor do Seguro
VLR_ENTR	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor da Entrada
DSP_ACES	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor das Despesas Acessórias
VLR_ACR_D SCT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor de Acréscimo ou desconto.... Este campo armazena valores positivos para acréscimos e valores negativos para Descontos
SUB_TOT	NUMERIC(18,4)	NUMERO_GRANDE_4CD	Valor do SubTotal do Pedido
TOT_PED	NUMERIC(18,4)	NUMERO_GRANDE_4CD	Valor Total do Pedido
QTD_VOL	SMALLINT	VALORS	Quantidade de Volumes
PES_LIQ	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Peso líquido dos Produtos
PES_BRU	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Peso bruto dos produtos
NUM_VOL	SMALLINT	VALORS	
TIPO_VOL	VARCHAR(20)	DESCRICAO_20	Tipo do Volume
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro

Name	Data Type	Domain	Comment
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.5.3 Dependências da Tabela PEDIDOS

Name	Code	Class Name
FK_PEDIDOS_PEDIDOS_ITENS	FK_PEDIDOS_PEDIDOS_ITENS	Reference
FK_VINC_PEDIDOS_PEDIDOS	FK_VINC_PEDIDOS_PEDIDOS	Reference
FK_VINCULO_PEDID_PEDIDOS_VI NC	FK_VINCULO_PEDID_PEDIDOS_VI NC	Reference
FK_PEDIDOS_ENTRE_PEDIDOS	FK_PEDIDOS_ENTRE_PEDIDOS	Reference
FK_PEDIDOS_HISTORICOS_PEDI DOS	FK_PEDIDOS_HISTORICOS_PEDI DOS	Reference
FK_PEDIDOS_MENSAGENS_PEDI DOS	FK_PEDIDOS_MENSAGENS_PEDI DOS	Reference
FK_PEDIDOS_DESCO_PEDIDOS	FK_PEDIDOS_DESCO_PEDIDOS	Reference
FK_PEDIDOS_PARCE_PEDIDOS	FK_PEDIDOS_PARCE_PEDIDOS	Reference
FK_DOCUMENTOS_PEDIDOS_PE DIDOS	FK_DOCUMENTOS_PEDIDOS_PE DIDOS	Reference
FK_CAPA_PEDIDOS_EMPRESAS	FK_CAPA_PEDIDOS_EMPRESAS	Reference
FK_PEDIDOS_CADASTROS	FK_PEDIDOS_CADASTROS	Reference
FK_PEDIDOS_TIPO_PAGAMENT	FK_PEDIDOS_TIPO_PAGAMENT	Reference
FK_PEDIDOS_TIPO_DESCONTO	FK_PEDIDOS_TIPO_DESCONTO	Reference
FK_PEDIDOS_TIPO_PEDIDOS	FK_PEDIDOS_TIPO_PEDIDOS	Reference
FK_PEDIDOS_VENDEDORES	FK_PEDIDOS_VENDEDORES	Reference
FK_PEDIDOS_TRANSPORTADOR AS	FK_PEDIDOS_TRANSPORTADOR AS	Reference
FK_PEDIDOS_TABELA_PRECOS	FK_PEDIDOS_TABELA_PRECOS	Reference
FK_PEDIDOS_TIPO_STATUS_PED IDOS	FK_PEDIDOS_TIPO_STATUS_PED IDOS	Reference
FK_PEDIDOS_TIPO_MOVIMENTO S	FK_PEDIDOS_TIPO_MOVIMENTO S	Reference
FK_PEDIDOS_TIPO_PRAZO_EN	FK_PEDIDOS_TIPO_PRAZO_EN	Reference
FK_PEDIDOS_FINALIZADORAS	FK_PEDIDOS_FINALIZADORAS	Reference
PEDIDOS	PEDIDOS	Shortcut of table

1.3.5.4 Lista de Chaves da Tabela PEDIDOS

Name	Code	Primary
PK_PEDIDOS	PK_PEDIDOS	X

1.3.5.5 Chave PK_PEDIDOS da Tabela PEDIDOS

1.3.5.5.1 Lista de Colunas da Chave PK_PEDIDOS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
PK_PEDIDOS	PK_PEDIDOS

1.3.5.5.2 Dependências da Chave PK_PEDIDOS

Name	Code	Class Name
PK_PEDIDOS	PK_PEDIDOS	Index
FK_PEDIDOS_PEDIDOS_ITENS	FK_PEDIDOS_PEDIDOS_ITENS	Reference
FK_VINC_PEDIDOS_PEDIDOS	FK_VINC_PEDIDOS_PEDIDOS	Reference
FK_VINCULO_PEDID_PEDIDOS_VI NC	FK_VINCULO_PEDID_PEDIDOS_VI NC	Reference
FK_PEDIDOS_ENTRE_PEDIDOS	FK_PEDIDOS_ENTRE_PEDIDOS	Reference
FK_PEDIDOS_HISTORICOS_PEDI DOS	FK_PEDIDOS_HISTORICOS_PEDI DOS	Reference
FK_PEDIDOS_MENSAGENS_PEDI DOS	FK_PEDIDOS_MENSAGENS_PEDI DOS	Reference
FK_PEDIDOS_DESCO_PEDIDOS	FK_PEDIDOS_DESCO_PEDIDOS	Reference
FK_PEDIDOS_PARCE_PEDIDOS	FK_PEDIDOS_PARCE_PEDIDOS	Reference
FK_DOCUMENTOS_PEDIDOS_PE DIDOS	FK_DOCUMENTOS_PEDIDOS_PE DIDOS	Reference

1.3.5.6 Lista de Índices PEDIDOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_PEDIDOS	PK_PEDIDOS	X		X		
IDX_PEDIDOS 01	IDX_PEDIDOS 01					
IDX_PEDIDOS 02	IDX_PEDIDOS 02					

1.3.5.7 Índice PK_PEDIDOS da Tabela PEDIDOS

1.3.5.7.1 Lista de Colunas do Índice PK_PEDIDOS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
PK_PEDIDOS	PK_PEDIDOS	Ascending

1.3.5.8 Índice IDX_PEDIDOS01 da Tabela PEDIDOS

1.3.5.8.1 Lista de Colunas do Índice IDX_PEDIDOS01

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_CADASTROS	FK_CADASTROS	Ascending
PK_PEDIDOS	PK_PEDIDOS	Ascending

1.3.5.9 Índice IDX_PEDIDOS02 da Tabela PEDIDOS**1.3.5.9.1 Lista de Colunas do Índice IDX_PEDIDOS02**

Name	Code	Sort
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_CADASTROS	FK_CADASTROS	Ascending
NUM_EXTR	NUM_EXTR	Ascending
FK_EMPRESAS	FK_EMPRESAS	Ascending

1.3.5.10 Lista de Triggers da Tabela PEDIDOS

Name	Code
TBU0_PEDIDOS	TBU0_PEDIDOS
TBIG_PEDIDOS	TBIG_PEDIDOS
TAU0_PEDIDOS	TAU0_PEDIDOS
TAI0_PEDIDOS	TAI0_PEDIDOS

1.3.6 Tabela PEDIDOS_DESCONTOS

1.3.6.1 Card of table PEDIDOS_DESCONTOS

Name	PEDIDOS_DESCONTOS
Code	PEDIDOS_DESCONTOS
Comment	Tabela que armazena todos os descontos dados pelo sistema e usuário
Primary Key	PK_PEDIDOS_DESCONTOS

1.3.6.2 Lista de Colunas da Tabela PEDIDOS_DESCONTOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
PK_PEDIDOS_DESCONTOS	SMALLINT	VALORS	Código
DSC_DSCT	VARCHAR(30)	DESCRICAO_30RQ	Descrição do desconto
IDX_DSCT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Índice do desconto. Aplicado no valor conforme a ordem e o flag que indica como deve ser calculado
FLAG_TDSC T	SMALLINT	FLAG_TIPO_DESCONTO	Flag que indica o tipo do desconto aplicado à uma operação 0 ==> Percentual 1 ==> Índice Multiplicador 2 ==> Índice Divisor 3 ==> Valor
FLAG_DSTQ	SMALLINT	FLAG_SIM_NAO	Flag que indica se o desconto será destacado na NF. Se o flag for setado os descontos farão parte do corpo da NF, caso contrário os descontos serão somente somados e o valor total destacado no campo Valor de acréscimo desconto.
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.6.3 Dependências da Tabela PEDIDOS_DESCONTOS

Name	Code	Class Name
FK_PEDIDOS_DESCO_PEDIDOS	FK_PEDIDOS_DESCO_PEDIDOS	Reference

1.3.6.4 Lista de Chaves da Tabela PEDIDOS_DESCONTOS

Name	Code	Primary
PK_PEDIDOS_DESCONTOS	PK_PEDIDOS_DESCONTOS	X

1.3.6.5 Chave PK_PEDIDOS_DESCONTOS da Tabela PEDIDOS_DESCONTOS**1.3.6.5.1 Lista de Colunas da Chave PK_PEDIDOS_DESCONTOS**

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
PK_PEDIDOS_DESCONTOS	PK_PEDIDOS_DESCONTOS

1.3.6.5.2 Dependências da Chave PK_PEDIDOS_DESCONTOS

Name	Code	Class Name
PK_PEDIDOS_DESCONTOS	PK_PEDIDOS_DESCONTOS	Index

1.3.6.6 Lista de Índices PEDIDOS_DESCONTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_DESCONTOS	PK_PEDIDOS_DESCONTOS	X		X		

1.3.6.7 Índice PK_PEDIDOS_DESCONTOS da Tabela PEDIDOS_DESCONTOS**1.3.6.7.1 Lista de Colunas do Índice PK_PEDIDOS_DESCONTOS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
PK_PEDIDOS_DESCONTOS	PK_PEDIDOS_DESCONTOS	Ascending

1.3.6.8 Lista de Triggers da Tabela PEDIDOS_DESCONTOS

Name	Code
TBI0_PEDIDOS_DESCONTOS	TBI0_PEDIDOS_DESCONTOS
TBU0_PEDIDOS_DESCONTOS	TBU0_PEDIDOS_DESCONTOS

1.3.7 Tabela PEDIDOS_ENTREGA

1.3.7.1 Card of table PEDIDOS_ENTREGA

Name	PEDIDOS_ENTREGA
Code	PEDIDOS_ENTREGA
Comment	tabela que armazena os endereços de entrega e/ou redespacho
Primary Key	PK_PEDIDOS_ENTREGA

1.3.7.2 Lista de Colunas da Tabela PEDIDOS_ENTREGA

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
UF_ENTR	CHAR(2)	UF	Estado em que será feita a entrega
MUN_ENTR	INTEGER	VALORI	Município de entrega do pedido
END_ENTR	BLOB SUB_TYPE 1 SEGMENT SIZE 80	BLOB_TEXTO	Endereço completo de entrega do pedido
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.7.3 Dependências da Tabela PEDIDOS_ENTREGA

Name	Code	Class Name
FK_PEDIDOS_ENTRE_PEDIDOS	FK_PEDIDOS_ENTRE_PEDIDOS	Reference

1.3.7.4 Lista de Chaves da Tabela PEDIDOS_ENTREGA

Name	Code	Primary
PK_PEDIDOS_ENTREGA	PK_PEDIDOS_ENTREGA	X

1.3.7.5 Chave PK_PEDIDOS_ENTREGA da Tabela PEDIDOS_ENTREGA

1.3.7.5.1 Lista de Colunas da Chave PK_PEDIDOS_ENTREGA

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS

Name	Code	Class Name
PK_PEDIDOS_ENTREGA	PK_PEDIDOS_ENTREGA	Index

1.3.7.6 Lista de Índices PEDIDOS_ENTREGA

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_ENTREGA	PK_PEDIDOS_ENTREGA	X		X		

1.3.7.7 Índice PK_PEDIDOS_ENTREGA da Tabela PEDIDOS_ENTREGA

1.3.7.7.1 Lista de Colunas do Índice PK_PEDIDOS_ENTREGA

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending

1.3.7.8 Lista de Triggers da Tabela PEDIDOS_ENTREGA

Name	Code
TBU0_PEDIDOS_ENTREGA	TBU0_PEDIDOS_ENTREGA

1.3.8 Tabela PEDIDOS_HISTORICOS

1.3.8.1 Card of table PEDIDOS_HISTORICOS

Name	PEDIDOS_HISTORICOS
Code	PEDIDOS_HISTORICOS
Comment	Tabela que armazena o histórico dos status do pedido. Esta tabela é usada para controlar se algum status já foi usado na tabela fazendo que a banxa de um status anterior não seja feita duas vezes...
Primary Key	PK_PEDIDOS_HISTORICOS

1.3.8.2 Lista de Colunas da Tabela PEDIDOS_HISTORICOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
PK_PEDIDOS_HISTORICOS	SMALLINT	VALORS	Código sequencial
FK_TIPO_STATUS_PEDIDOS	SMALLINT	VALORS	Código do Status do Pedido
FLAG_BXASST	SMALLINT	FLAG_SIM_NAO	Flag que indica se o status teve modificações do estoque
DSC_HIST	VARCHAR(100)	DESCRICAO_100RQ	Descrição do Histórico
DTHR_HIST	TIMESTAMP	DATA_HORA_DEF	Data e hora de criação do histórico
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.8.3 Dependências da Tabela PEDIDOS_HISTORICOS

Name	Code	Class Name
FK_PEDIDOS_HISTORICOS_PEDIDOS	FK_PEDIDOS_HISTORICOS_PEDIDOS	Reference
FK_PEDIDOS_STT_H_TIPO_STATUS_P	FK_PEDIDOS_STT_H_TIPO_STATUS_P	Reference

1.3.8.4 Lista de Chaves da Tabela PEDIDOS_HISTORICOS

Name	Code	Primary
PK_PEDIDOS_HISTORICOS	PK_PEDIDOS_HISTORICOS	X

1.3.8.5 Chave PK_PEDIDOS_HISTORICOS da Tabela PEDIDOS_HISTORICOS**1.3.8.5.1 Lista de Colunas da Chave PK_PEDIDOS_HISTORICOS**

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
PK_PEDIDOS_HISTORICOS	PK_PEDIDOS_HISTORICOS

1.3.8.5.2 Dependências da Chave PK_PEDIDOS_HISTORICOS

Name	Code	Class Name
PK_PEDIDOS_HISTORICOS	PK_PEDIDOS_HISTORICOS	Index

1.3.8.6 Lista de Índices PEDIDOS_HISTORICOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_HISTORICOS	PK_PEDIDOS_HISTORICOS	X		X		

1.3.8.7 Índice PK_PEDIDOS_HISTORICOS da Tabela PEDIDOS_HISTORICOS**1.3.8.7.1 Lista de Colunas do Índice PK_PEDIDOS_HISTORICOS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
PK_PEDIDOS_HISTORICOS	PK_PEDIDOS_HISTORICOS	Ascending

1.3.8.8 Lista de Triggers da Tabela PEDIDOS_HISTORICOS

Name	Code
TBU0_PEDIDOS_HISTORICOS	TBU0_PEDIDOS_HISTORICOS
TAIG_PEDIDOS_HISTORICOS	TAIG_PEDIDOS_HISTORICOS

1.3.9 Tabela PEDIDOS_ITENS

1.3.9.1 Card of table PEDIDOS_ITENS

Name	PEDIDOS_ITENS
Code	PEDIDOS_ITENS
Comment	Tabela que armazena os itens dos pedidos....
Primary Key	PK_PEDIDOS_ITENS

1.3.9.2 Lista de Colunas da Tabela PEDIDOS_ITENS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
PK_PEDIDOS_ITENS	SMALLINT	VALORS	Número do Ítem do Pedido
FK_TABELA_PRECOS	SMALLINT	VALORS	
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Grupo de Movimentos
FK_TIPO_MOVIMENTOS	SMALLINT	VALORS	Referência à tabela para buscar a Natureza da Operação do item
FK_UNIDADES	SMALLINT	VALORS	Unidade do Ítem
FK_CARGAS_PRODUCAO	INTEGER	VALORI	Código da Carga
FK_PRODUTOS	INTEGER	VALORI	Código do Produto do Pedido
FK_CLASSIFICACAO	INTEGER	VALORI	Código do Plano de Contas (Sequencial com DV)
FK_ALMOXARIFADOS	SMALLINT	VALORS	Código do Almojarifado
FK_LOTACOES	SMALLINT	VALORS	Código da lotação
REF_PRODUTO	VARCHAR(20)	CODIGO_PRODUTO	Referência do Produto
LOT_ITEM	VARCHAR(10)	DESCRICAO_10	Lotação do Ítem
QTD_ITEM	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Quantidade do Ítem
VLR_TAB	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor Unitário da tabela de preços
VLR_UNIT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor Unitário
SUB_TOT	NUMERIC(11,4)	NUMERO_MEDIO_4CD	
VLR_ACR_D SCT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor do desconto calculado
TOT_ITEM	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Total do Ítem

Name	Data Type	Domain	Comment
SIT_TRIB	CHAR(3)	SITUACAO_TRIBUTARIA	Situação Tributária do Ítem
ALQ_ISS	NUMERIC(5,2)	NUMERO_PERCENTUAL	Alíquota de iss sobre o produto
ALQ_ICMS	NUMERIC(5,2)	NUMERO_PERCENTUAL	
ALQ_ICMSS	NUMERIC(5,2)	NUMERO_PERCENTUAL	Alíquota do ICMS Substituição
ALQ_IPI	NUMERIC(5,2)	NUMERO_PERCENTUAL	Percentual de Ipi sobre o Ítem
ALQ_OTR	NUMERIC(5,2)	NUMERO_PERCENTUAL	Alíquota de outro imposto
COD_ISS_E CF	CHAR(3)	CODIGO_IMPOSTO_ECF	Código do iss na impressora fiscal
COD_ICMS_ ECF	CHAR(3)	CODIGO_IMPOSTO_ECF	Código do Imposto na Impressora de Cupom Fiscal
COD_ICMSS_ ECF	CHAR(3)	CODIGO_IMPOSTO_ECF	Código do ICMS Substituição na Impressora Fiscal
COD_IPI_EC F	CHAR(3)	CODIGO_IMPOSTO_ECF	Código do IPI na impressora fiscal
COD_OTR_ ECF	SMALLINT	VALORS	Códio de outra alíquota de imposto na impressora fiscal
FLAG_PRM	SMALLINT	FLAG_SIM_NAO	Flag que indica se o ítem está em promoção ou não
FLAG_PEND	SMALLINT	FLAG_SIM_NAO	Flag que indica se o Ítem possui pendências
FLAG_BXAS TT	SMALLINT	FLAG_SIM_NAO	Flag que indica se será feita a baixa de estoques nos itens para o status atual
FLAG_SRV	SMALLINT	FLAG_SIM_NAO	Flag que indica se o produtoo é um serviço
FLAG_CONF	SMALLINT	FLAG_SIM_NAO	Flag que indica se existe uma configuração para o produto
DTA_FAT	DATE	DATA	Data de faturamento do pedido
DTA_LIQ	DATE	DATA	Data da Liquidação do Ítem do pedido
NUM_EXT	INTEGER	VALORI	Campo usado para armazenar um número de pedido externo
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro

1.3.9.3 Dependências da Tabela PEDIDOS_ITENS

Name	Code	Class Name
FK_PENDENCIAS_PE_ITENS_PED IDOS	FK_PENDENCIAS_PE_ITENS_PED IDOS	Reference
FK_PEDIDOS_ITENS_PEDIDOS_IT ENS	FK_PEDIDOS_ITENS_PEDIDOS_IT ENS	Reference
FK_PEDIDOS_PEDIDOS_ITENS	FK_PEDIDOS_PEDIDOS_ITENS	Reference

Name	Code	Class Name
FK_PEDIDOS_ITENS_UNIDADES	FK_PEDIDOS_ITENS_UNIDADES	Reference
FK_PEDIDOS_ITENS_TIPO_MOVIMENT	FK_PEDIDOS_ITENS_TIPO_MOVIMENT	Reference
FK_PEDIDOS_ITENS_TABELA_PRECOS	FK_PEDIDOS_ITENS_TABELA_PRECOS	Reference
FK_PEDIDOS_ITENS_CARGAS_PRODUC	FK_PEDIDOS_ITENS_CARGAS_PRODUC	Reference
FK_PEDIDOS_ITENS_PRODUTOS	FK_PEDIDOS_ITENS_PRODUTOS	Reference
FK_PEDIDOS_ITENS_CLASSIFICACAO	FK_PEDIDOS_ITENS_CLASSIFICACAO	Reference
FK_PEDIDOS_ITENS_LOTACOES	FK_PEDIDOS_ITENS_LOTACOES	Reference

1.3.9.4 Lista de Chaves da Tabela PEDIDOS_ITENS

Name	Code	Primary
PK_PEDIDOS_ITENS	PK_PEDIDOS_ITENS	X

1.3.9.5 Chave PK_PEDIDOS_ITENS da Tabela PEDIDOS_ITENS

1.3.9.5.1 Lista de Colunas da Chave PK_PEDIDOS_ITENS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
PK_PEDIDOS_ITENS	PK_PEDIDOS_ITENS

1.3.9.5.2 Dependências da Chave PK_PEDIDOS_ITENS

Name	Code	Class Name
PK_PEDIDOS_ITENS	PK_PEDIDOS_ITENS	Index
FK_PENDENCIAS_PE_ITENS_PEDIDOS	FK_PENDENCIAS_PE_ITENS_PEDIDOS	Reference
FK_PEDIDOS_ITENS_PEDIDOS_ITENS	FK_PEDIDOS_ITENS_PEDIDOS_ITENS	Reference

1.3.9.6 Lista de Índices PEDIDOS_ITENS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_ITENS	PK_PEDIDOS_ITENS	X		X		

1.3.9.7 Índice PK_PEDIDOS_ITENS da Tabela PEDIDOS_ITENS

1.3.9.7.1 Lista de Colunas do Índice PK_PEDIDOS_ITENS

Name	Code	Sort
------	------	------

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
PK_PEDIDOS_ITENS	PK_PEDIDOS_ITENS	Ascending

1.3.9.8 Lista de Triggers da Tabela PEDIDOS_ITENS

Name	Code
TBIG_PEDIDOS_ITENS	TBIG_PEDIDOS_ITENS
TBU0_PEDIDOS_ITENS	TBU0_PEDIDOS_ITENS

1.3.10 Tabela PEDIDOS_ITENS_CONF

1.3.10.1 Card of table PEDIDOS ITENS CONF

Name	PEDIDOS_ITENS_CONF
Code	PEDIDOS_ITENS_CONF
Comment	Tabela que armazena as configurações dos Ítens do Pedido. Tecido; Cor; etc...
Primary Key	PK_PEDIDOS_ITENS_CONF

1.3.10.2 Lista de Colunas da Tabela PEDIDOS_ITENS_CONF

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
FK_PEDIDOS_ITENS	SMALLINT	VALORS	Número do Ítem do Pedido
PK_PEDIDOS_ITENS_CONF	SMALLINT	VALORS	Chave seqüencial
FK_PRODUTOS	INTEGER	VALORI	Código do Produto
FK_COMPONENTES	SMALLINT	VALORS	Código do Tipo de Componente
FK_ACABAMENTOS	SMALLINT	VALORS	Código do Tipo de Acabamento
FK_TIPO_REFERENCIAS	SMALLINT	VALORS	Código da Referência
FK_INSUMOS	INTEGER	VALORI	Código do insumo utilizado no acabamento
COD_REF	VARCHAR(20)	CODIGO_PRODUTO	Código da Referência ao insumo / Composição que fará o acabamento do produto.
VLR_ITM	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Valor do Ítem
QTD_COMP	SMALLINT	VALORS	Quantidade de componentes
QTD_COMP_TOT	SMALLINT	VALORS	
QTD_INS	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Quantidade de Insumo para o componente
FLAG_FRNC LI	SMALLINT	FLAG_SIM_NAO	Flag que indica se o acabamento eh fornecido pelo cliente
FLAG_PEND	SMALLINT	FLAG_SIM_NAO	Flag que indica a pendência do Acabamento aguardando o fornecimento do Cliente

Name	Data Type	Domain	Comment
FLAG_CNTR	SMALLINT	FLAG_SIM_NAO	Flag que indica se o tecido deve ser centralizado no móvel
FLAG_BXAS TT	SMALLINT	FLAG_SIM_NAO	Flag que indica se baixa de estoques será efetuada para o status atual
FK_TIPO_D OCUMENTO S	SMALLINT	VALORS	Tipo do Documento que fez a liberação do acabamento
NUM_DOC_ LIB	INTEGER	VALORI	Número do documento que liberou o acabamento
PER_DSCT_ INS	NUMERIC(5,2)	NUMERO_PERCENTUAL	Percentual de desconto usado no insumo que está em promoção
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.10.3 Dependências da Tabela PEDIDOS_ITENS_CONF

Name	Code	Class Name
FK_PEDIDOS_ITENS_PEDIDOS_IT ENS	FK_PEDIDOS_ITENS_PEDIDOS_IT ENS	Reference
FK_PEDIDOS_ITENS_CONF_PRO DUTOS	FK_PEDIDOS_ITENS_CONF_PRO DUTOS	Reference
FK_PEDIDOS_ITENS_REFERENCI AS	FK_PEDIDOS_ITENS_REFERENCI AS	Reference

1.3.10.4 Lista de Chaves da Tabela PEDIDOS_ITENS_CONF

Name	Code	Primary
PK_PEDIDOS_ITENS_CONF	PK_PEDIDOS_ITENS_CONF	X

1.3.10.5 Chave PK_PEDIDOS_ITENS_CONF da Tabela PEDIDOS_ITENS_CONF

1.3.10.5.1 Lista de Colunas da Chave PK_PEDIDOS_ITENS_CONF

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
FK_PEDIDOS_ITENS	FK_PEDIDOS_ITENS
PK_PEDIDOS_ITENS_CONF	PK_PEDIDOS_ITENS_CONF

1.3.10.5.2 Dependências da Chave PK_PEDIDOS_ITENS_CONF

Name	Code	Class Name
------	------	------------

Name	Code	Class Name
PK_PEDIDOS_ITENS_CONF	PK_PEDIDOS_ITENS_CONF	Index

1.3.10.6 Lista de Índices PEDIDOS_ITENS_CONF

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_ITENS_CONF	PK_PEDIDOS_ITENS_CONF	X		X		

1.3.10.7 Índice PK_PEDIDOS_ITENS_CONF da Tabela PEDIDOS_ITENS_CONF

1.3.10.7.1 Lista de Colunas do Índice PK_PEDIDOS_ITENS_CONF

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
FK_PEDIDOS_ITENS	FK_PEDIDOS_ITENS	Ascending
PK_PEDIDOS_ITENS_CONF	PK_PEDIDOS_ITENS_CONF	Ascending

1.3.10.8 Lista de Triggers da Tabela PEDIDOS_ITENS_CONF

Name	Code
TBU0_PEDIDOS_ITENS_CONF	TBU0_PEDIDOS_ITENS_CONF
TBIG_PEDIDOS_ITENS_CONF	TBIG_PEDIDOS_ITENS_CONF

1.3.11 Tabela PEDIDOS_MENSAGENS

1.3.11.1 Card of table PEDIDOS_MENSAGENS

Name	PEDIDOS_MENSAGENS
Code	PEDIDOS_MENSAGENS
Comment	
Primary Key	PK_PEDIDOS_MENSAGENS

1.3.11.2 Lista de Colunas da Tabela PEDIDOS_MENSAGENS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
PK_PEDIDOS_MENSAGENS	INTEGER	VALORI	código da Mensagem
FK_DEPARTAMENTOS	SMALLINT	VALORS	Código do Departamento
DTHR_MSG	TIMESTAMP	DATA_HORA_DEF	Data e Hora da Mensagem
DTHR_RCBM	TIMESTAMP	DATA_HORA	Data e hora do recebimento
TEXT_MSG	BLOB SUB_TYPE 1 SEGMENT SIZE 80	BLOB_TEXTO	Texto da Mensagem
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.11.3 Dependências da Tabela PEDIDOS_MENSAGENS

Name	Code	Class Name
FK_PEDIDOS_MENSAGENS_PEDIDOS	FK_PEDIDOS_MENSAGENS_PEDIDOS	Reference
FK_PEDIDOS_MENSA_DEPARTAMENTOS	FK_PEDIDOS_MENSA_DEPARTAMENTOS	Reference

1.3.11.4 Lista de Chaves da Tabela PEDIDOS_MENSAGENS

Name	Code	Primary
PK_PEDIDOS_MENSAGENS	PK_PEDIDOS_MENSAGENS	X

1.3.11.5 Chave PK_PEDIDOS_MENSAGENS da Tabela PEDIDOS_MENSAGENS

1.3.11.5.1 Lista de Colunas da Chave PK_PEDIDOS_MENSAGENS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
PK_PEDIDOS_MENSAGENS	PK_PEDIDOS_MENSAGENS

1.3.11.5.2 Dependências da Chave PK_PEDIDOS_MENSAGENS

Name	Code	Class Name
PK_PEDIDOS_MENSAGENS	PK_PEDIDOS_MENSAGENS	Index

1.3.11.6 Lista de Índices PEDIDOS_MENSAGENS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_MENSAGENS	PK_PEDIDOS_MENSAGENS	X		X		

1.3.11.7 Índice PK_PEDIDOS_MENSAGENS da Tabela PEDIDOS_MENSAGENS**1.3.11.7.1 Lista de Colunas do Índice PK_PEDIDOS_MENSAGENS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
PK_PEDIDOS_MENSAGENS	PK_PEDIDOS_MENSAGENS	Ascending

1.3.11.8 Lista de Triggers da Tabela PEDIDOS_MENSAGENS

Name	Code
TBU0_PEDIDOS_MENSAGENS	TBU0_PEDIDOS_MENSAGENS
TBIG_PEDIDOS_MENSAGENS	TBIG_PEDIDOS_MENSAGENS

1.3.12 Tabela PEDIDOS_PARCELAS

1.3.12.1 Card of table PEDIDOS_PARCELAS

Name	PEDIDOS_PARCELAS
Code	PEDIDOS_PARCELAS
Comment	Tabela que armazena as parcelas geradas para o faturamento previsto do pedido
Primary Key	PK_PEDIDOS_PARCELAS

1.3.12.2 Lista de Colunas da Tabela PEDIDOS_PARCELAS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
PK_PEDIDOS_PARCELAS	DATE	DATA	
NUM_PARC	SMALLINT	VALORS	Número da Parcela
VLR_PARC	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Valor da parcela
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.12.3 Dependências da Tabela PEDIDOS_PARCELAS

Name	Code	Class Name
FK_PEDIDOS_PARCE_PEDIDOS	FK_PEDIDOS_PARCE_PEDIDOS	Reference

1.3.12.4 Lista de Chaves da Tabela PEDIDOS_PARCELAS

Name	Code	Primary
PK_PEDIDOS_PARCELAS	PK_PEDIDOS_PARCELAS	X

1.3.12.5 Chave PK_PEDIDOS_PARCELAS da Tabela PEDIDOS_PARCELAS

1.3.12.5.1 Lista de Colunas da Chave PK_PEDIDOS_PARCELAS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS
FK_PEDIDOS	FK_PEDIDOS
PK_PEDIDOS_PARCELAS	PK_PEDIDOS_PARCELAS

1.3.12.5.2 Dependências da Chave PK_PEDIDOS_PARCELAS

Name	Code	Class Name
PK_PEDIDOS_PARCELAS	PK_PEDIDOS_PARCELAS	Index

1.3.12.6 Lista de Índices PEDIDOS_PARCELAS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_PARCELAS	PK_PEDIDOS_PARCELAS	X		X		

1.3.12.7 Índice PK_PEDIDOS_PARCELAS da Tabela PEDIDOS_PARCELAS**1.3.12.7.1 Lista de Colunas do Índice PK_PEDIDOS_PARCELAS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
PK_PEDIDOS_PARCELAS	PK_PEDIDOS_PARCELAS	Ascending

1.3.12.8 Lista de Triggers da Tabela PEDIDOS_PARCELAS

Name	Code
TBU0_PEDIDOS_PARCELAS	TBU0_PEDIDOS_PARCELAS

1.3.13 Tabela PEDIDOS_PENDENCIAS

1.3.13.1 Card of table PEDIDOS_PENDENCIAS

Name	PEDIDOS_PENDENCIAS
Code	PEDIDOS_PENDENCIAS
Comment	Tabela que armazena uma estatística de produtos que foram pedidos e não encontrados em estoques será gerada pela procedure STP_BUSCA_PRODUTOS e STP_BUSCA_CONFIGURACOES
Primary Key	PK_PEDIDOS_PENDENCIAS

1.3.13.2 Lista de Colunas da Tabela PEDIDOS_PENDENCIAS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
FK_PEDIDOS_ITEMS	SMALLINT	VALORS	Número do Ítem do Pedido
FK_PRODUTOS	INTEGER	VALORI	Código do Produto ou Insumo do Pedido
QTD_PEND	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Quantidade da Pendência do Ítem
VLR_UNIT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor Unitário do Ítem do Pedido
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.13.3 Dependências da Tabela PEDIDOS_PENDENCIAS

Name	Code	Class Name
FK_PENDENCIAS_PE_ITENS_PEDIDOS	FK_PENDENCIAS_PE_ITENS_PEDIDOS	Reference

1.3.13.4 Lista de Chaves da Tabela PEDIDOS_PENDENCIAS

Name	Code	Primary
PK_PEDIDOS_PENDENCIAS	PK_PEDIDOS_PENDENCIAS	X

1.3.13.5 Chave PK_PEDIDOS_PENDENCIAS da Tabela PEDIDOS_PENDENCIAS

1.3.13.5.1 Lista de Colunas da Chave PK_PEDIDOS_PENDENCIAS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS

Name	Code
FK_PEDIDOS	FK_PEDIDOS
FK_PEDIDOS_ITEMS	FK_PEDIDOS_ITEMS
FK_PRODUTOS	FK_PRODUTOS

1.3.13.5.2 Dependências da Chave PK_PEDIDOS_PENDENCIAS

Name	Code	Class Name
PK_PEDIDOS_PENDENCIAS	PK_PEDIDOS_PENDENCIAS	Index

1.3.13.6 Lista de Índices PEDIDOS_PENDENCIAS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_PENDENCIAS	PK_PEDIDOS_PENDENCIAS	X		X		

1.3.13.7 Índice PK_PEDIDOS_PENDENCIAS da Tabela PEDIDOS_PENDENCIAS

1.3.13.7.1 Lista de Colunas do Índice PK_PEDIDOS_PENDENCIAS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
FK_PEDIDOS_ITEMS	FK_PEDIDOS_ITEMS	Ascending
FK_PRODUTOS	FK_PRODUTOS	Ascending

1.3.13.8 Lista de Triggers da Tabela PEDIDOS_PENDENCIAS

Name	Code
TBU0_PEDIDOS_PENDENCIAS	TBU0_PEDIDOS_PENDENCIAS

1.3.14 Tabela PEDIDOS_VINCULOS

1.3.14.1 Card of table PEDIDOS_VINCULOS

Name	PEDIDOS_VINCULOS
Code	PEDIDOS_VINCULOS
Comment	
Primary Key	PK_PEDIDOS_VINCULOS

1.3.14.2 Lista de Colunas da Tabela PEDIDOS_VINCULOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do pedido
FK_PEDIDOS	INTEGER	VALORI	Número do Pedido
FK_EMPRESAS_VINC	INTEGER	VALORI	Código da empresa do pedido vinculado
FK_TIPO_PEDIDOS_VINC	SMALLINT	VALORS	Código do tipo do pedido vinculado
FK_PEDIDOS_VINC	INTEGER	VALORI	Código do pedido vinculado
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.14.3 Dependências da Tabela PEDIDOS_VINCULOS

Name	Code	Class Name
FK_VINC_PEDIDOS_PEDIDOS	FK_VINC_PEDIDOS_PEDIDOS	Reference
FK_VINCULO_PEDID_PEDIDOS_VINC	FK_VINCULO_PEDID_PEDIDOS_VINC	Reference

1.3.14.4 Lista de Chaves da Tabela PEDIDOS_VINCULOS

Name	Code	Primary
PK_PEDIDOS_VINCULOS	PK_PEDIDOS_VINCULOS	X

1.3.14.5 Chave PK_PEDIDOS_VINCULOS da Tabela PEDIDOS_VINCULOS

1.3.14.5.1 Lista de Colunas da Chave PK_PEDIDOS_VINCULOS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS

Name	Code
FK_PEDIDOS	FK_PEDIDOS
FK_EMPRESAS_VINC	FK_EMPRESAS_VINC
FK_TIPO_PEDIDOS_VINC	FK_TIPO_PEDIDOS_VINC
FK_PEDIDOS_VINC	FK_PEDIDOS_VINC

1.3.14.5.2 Dependências da Chave PK_PEDIDOS_VINCULOS

Name	Code	Class Name
PK_PEDIDOS_VINCULOS	PK_PEDIDOS_VINCULOS	Index

1.3.14.6 Lista de Índices PEDIDOS_VINCULOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_PEDIDOS_VINCULOS	PK_PEDIDOS_VINCULOS	X		X		

1.3.14.7 Índice PK_PEDIDOS_VINCULOS da Tabela PEDIDOS_VINCULOS

1.3.14.7.1 Lista de Colunas do Índice PK_PEDIDOS_VINCULOS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_TIPO_PEDIDOS	FK_TIPO_PEDIDOS	Ascending
FK_PEDIDOS	FK_PEDIDOS	Ascending
FK_EMPRESAS_VINC	FK_EMPRESAS_VINC	Ascending
FK_TIPO_PEDIDOS_VINC	FK_TIPO_PEDIDOS_VINC	Ascending
FK_PEDIDOS_VINC	FK_PEDIDOS_VINC	Ascending

1.3.14.8 Lista de Triggers da Tabela PEDIDOS_VINCULOS

Name	Code
TBU0_PEDIDOS_VINCULOS	TBU0_PEDIDOS_VINCULOS

1.3.15 Tabela TIPO_DESCONTOS

1.3.15.1 Card of table TIPO_DESCONTOS

Name	TIPO_DESCONTOS
Code	TIPO_DESCONTOS
Comment	Tabela que informa o tipo de desconto do fornecedor
Primary Key	PK_TIPO_DESCONTOS

1.3.15.2 Lista de Colunas da Tabela TIPO_DESCONTOS

Name	Data Type	Domain	Comment
PK_TIPO_D ESCONTOS	SMALLINT	VALORS	Código do tipo do desconto
FK_GRUPO S_MOVIMEN TOS	SMALLINT	VALORS	Código do Tipo de Movimento
DSC_TDSCT	VARCHAR(30)	DESCRICAO_30RQ	Descrição do tipo de descontos
KC_DESCO NTOS	SMALLINT	VALORS	Key Control da tabela descontos
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.15.3 Dependências da Tabela TIPO_DESCONTOS

Name	Code	Class Name
FK_DESCONTOS_TIPO_DESCON TO	FK_DESCONTOS_TIPO_DESCON TO	Reference
FK_PEDIDOS_TIPO_DESCONTO	FK_PEDIDOS_TIPO_DESCONTO	Reference
FK_TIPO_PAGAMENT_TIPO_DES CONTO	FK_TIPO_PAGAMENT_TIPO_DES CONTO	Reference
FK_CLIENTES_TIPO_DESCONTO S	FK_CLIENTES_TIPO_DESCONTO S	Reference
FK_PRODUTOS_FORN_TIPO_DES CONTO	FK_PRODUTOS_FORN_TIPO_DES CONTO	Reference
FK_DOCUMENTOS_TIPO_DESCO NTO	FK_DOCUMENTOS_TIPO_DESCO NTO	Reference
FK_TIPO_DESCONTO_GRUPOS_ MOVIME	FK_TIPO_DESCONTO_GRUPOS_ MOVIME	Reference
TIPO_DESCONTOS	TIPO_DESCONTOS	Shortcut of table
TIPO_DESCONTOS	TIPO_DESCONTOS	Shortcut of table

Name	Code	Class Name
TIPO_DESCONTOS	TIPO_DESCONTOS	Shortcut of table

1.3.15.4 Lista de Chaves da Tabela TIPO_DESCONTOS

Name	Code	Primary
PK_TIPO_DESCONTOS	PK_TIPO_DESCONTOS	X

1.3.15.5 Chave PK_TIPO_DESCONTOS da Tabela TIPO_DESCONTOS

1.3.15.5.1 Lista de Colunas da Chave PK_TIPO_DESCONTOS

Name	Code
PK_TIPO_DESCONTOS	PK_TIPO_DESCONTOS

1.3.15.5.2 Dependências da Chave PK_TIPO_DESCONTOS

Name	Code	Class Name
PK_TIPO_DESCONTOS	PK_TIPO_DESCONTOS	Index
FK_CLIENTES_TIPO_DESCONTOS	FK_CLIENTES_TIPO_DESCONTOS	Reference
FK_PRODUTOS_FORN_TIPO_DESCONTO	FK_PRODUTOS_FORN_TIPO_DESCONTO	Reference
FK_DESCONTOS_TIPO_DESCONTO	FK_DESCONTOS_TIPO_DESCONTO	Reference
FK_PEDIDOS_TIPO_DESCONTO	FK_PEDIDOS_TIPO_DESCONTO	Reference
FK_TIPO_PAGAMENT_TIPO_DESCONTO	FK_TIPO_PAGAMENT_TIPO_DESCONTO	Reference
FK_DOCUMENTOS_TIPO_DESCONTO	FK_DOCUMENTOS_TIPO_DESCONTO	Reference

1.3.15.6 Lista de Índices TIPO_DESCONTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_DESCONTOS	PK_TIPO_DESCONTOS	X		X		

1.3.15.7 Índice PK_TIPO_DESCONTOS da Tabela TIPO_DESCONTOS

1.3.15.7.1 Lista de Colunas do Índice PK_TIPO_DESCONTOS

Name	Code	Sort
PK_TIPO_DESCONTOS	PK_TIPO_DESCONTOS	Ascending

1.3.15.8 Lista de Triggers da Tabela TIPO_DESCONTOS

Name	Code
TBIG_TIPO_DESCONTOS	TBIG_TIPO_DESCONTOS

Name	Code
TBU0_TIPO_DESCONTOS	TBU0_TIPO_DESCONTOS

1.3.16 Tabela TIPO_FRETES

1.3.16.1 Card of table TIPO_FRETES

Name	TIPO_FRETES
Code	TIPO_FRETES
Comment	Tabela que Armazena os tipos de Fretes do Sistema
Primary Key	PK_TIPO_FRETES

1.3.16.2 Lista de Colunas da Tabela TIPO_FRETES

Name	Data Type	Domain	Comment
PK_TIPO_FRETES	SMALLINT	VALORS	Código dos Tipos de Fretes
FK_TIPO_PAGAMENTOS	SMALLINT	VALORS	Código do Tipo de Pagamento
FK_CLASSIFICACAO	SMALLINT	VALORS	Código do Grupo Financeiro
DSC_TPFRE	VARCHAR(30)	DESCRICAO_30RQ	Descrição do Tipo de Frete
FLAG_TP_FRE	SMALLINT	FLAG_TIPO_FRETE	Flag que indica qual o tipo do Frete 1 ==> Frete pago 2 ==> Frete a pagar 3 ==> Outros
FLAG_FIN	SMALLINT	FLAG_SIM_NAO	Flag que indica se este tipo de frete gera financeiro
FLAG_ACU	SMALLINT	FLAG_SIM_NAO	Flag que indica se o valor do frete será acumulado para cobrança posterior
FLAG_NF	SMALLINT	FLAG_SIM_NAO	Flag que indica se o valor do frete vai destacado na Nota Fiscal.....
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.16.3 Dependências da Tabela TIPO_FRETES

Name	Code	Class Name
FK_TIPOS_FRETES_TIPO_PAGAMENT	FK_TIPOS_FRETES_TIPO_PAGAMENT	Reference
TIPO_FRETES	TIPO_FRETES	Shortcut of table

1.3.16.4 Lista de Chaves da Tabela TIPO_FRETES

Name	Code	Primary
PK_TIPO_FRETES	PK_TIPO_FRETES	X

1.3.16.5 Chave PK_TIPO_FRETES da Tabela TIPO_FRETES**1.3.16.5.1 Lista de Colunas da Chave PK_TIPO_FRETES**

Name	Code
PK_TIPO_FRETES	PK_TIPO_FRETES

1.3.16.5.2 Dependências da Chave PK_TIPO_FRETES

Name	Code	Class Name
PK_TIPO_FRETES	PK_TIPO_FRETES	Index

1.3.16.6 Lista de Índices TIPO_FRETES

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_FRETES	PK_TIPO_FRETES	X		X		

1.3.16.7 Índice PK_TIPO_FRETES da Tabela TIPO_FRETES**1.3.16.7.1 Lista de Colunas do Índice PK_TIPO_FRETES**

Name	Code	Sort
PK_TIPO_FRETES	PK_TIPO_FRETES	Ascending

1.3.16.8 Lista de Triggers da Tabela TIPO_FRETES

Name	Code
TBIG_TIPO_FRETES	TBIG_TIPO_FRETES
TBU0_TIPO_FRETES	TBU0_TIPO_FRETES

1.3.17 Tabela TIPO_MOVIMENTOS

1.3.17.1 Card of table TIPO_MOVIMENTOS

Name	TIPO_MOVIMENTOS
Code	TIPO_MOVIMENTOS
Comment	Tabela que identifica qual vai ser a Natureza de operação dos documentos fiscais....
Primary Key	PK_TIPO_MOVIMENTOS

1.3.17.2 Lista de Colunas da Tabela TIPO_MOVIMENTOS

Name	Data Type	Domain	Comment
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Grupo de Movimentos
PK_TIPO_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
FK_CLASSIFICACAO	SMALLINT	VALORS	Código da Conta de Classificação do movimento
DSC_TMOV	VARCHAR(30)	DESCRICAO_30RQ	Descrição do Tipo de Movimento
FLAG_EXP	SMALLINT	FLAG_SIM_NAO	Flag que indica se o movimento é de exportação
FLAG_CNS	SMALLINT	FLAG_SIM_NAO	Flag que indica que o movimento é para consumidor final
FLAG_LDV	SMALLINT	FLAG_SIM_NAO	Flag que indica que a digitação do campo Valor unitário fica liberado a qualquer usuário.
NAT_OPE_D E	VARCHAR(5)	NATUREZA_DA_OPERACAO	Código da Natureza da Operação para Dentro do Estados
NAT_OPE_F E	VARCHAR(5)	NATUREZA_DA_OPERACAO	campo que discrimina a natureza de operações Fora do Estado
NAT_OPE_E X	VARCHAR(5)	NATUREZA_DA_OPERACAO	Código da Natureza da Operação Para Exportação
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Operador que Incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.17.3 Dependências da Tabela TIPO_MOVIMENTOS

Name	Code	Class Name
FK_PEDIDOS_ITENS_TIPO_MOVIMENT	FK_PEDIDOS_ITENS_TIPO_MOVIMENT	Reference
FK_PEDIDOS_TIPO_MOVIMENTOS	FK_PEDIDOS_TIPO_MOVIMENTOS	Reference
FK_ORDENS_SERVIC_TIPO_MOVIMENT	FK_ORDENS_SERVIC_TIPO_MOVIMENT	Reference

Name	Code	Class Name
FK_TIPO_MOVIMENT_GRUPOS_MOVIME	FK_TIPO_MOVIMENT_GRUPOS_MOVIME	Reference
TIPO_MOVIMENTOS	TIPO_MOVIMENTOS	Shortcut of table
TIPO_MOVIMENTOS	TIPO_MOVIMENTOS	Shortcut of table

1.3.17.4 Lista de Chaves da Tabela TIPO_MOVIMENTOS

Name	Code	Primary
PK_TIPO_MOVIMENTOS	PK_TIPO_MOVIMENTOS	X

1.3.17.5 Chave PK_TIPO_MOVIMENTOS da Tabela TIPO_MOVIMENTOS

1.3.17.5.1 Lista de Colunas da Chave PK_TIPO_MOVIMENTOS

Name	Code
FK_GRUPOS_MOVIMENTOS	FK_GRUPOS_MOVIMENTOS
PK_TIPO_MOVIMENTOS	PK_TIPO_MOVIMENTOS

1.3.17.5.2 Dependências da Chave PK_TIPO_MOVIMENTOS

Name	Code	Class Name
PK_TIPO_MOVIMENTOS	PK_TIPO_MOVIMENTOS	Index
FK_PEDIDOS_ITENS_TIPO_MOVIMENT	FK_PEDIDOS_ITENS_TIPO_MOVIMENT	Reference
FK_PEDIDOS_TIPO_MOVIMENTOS	FK_PEDIDOS_TIPO_MOVIMENTOS	Reference
FK_ORDENS_SERVIC_TIPO_MOVIMENT	FK_ORDENS_SERVIC_TIPO_MOVIMENT	Reference

1.3.17.6 Lista de Índices TIPO_MOVIMENTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_MOVIMENTOS	PK_TIPO_MOVIMENTOS	X		X		

1.3.17.7 Índice PK_TIPO_MOVIMENTOS da Tabela TIPO_MOVIMENTOS

1.3.17.7.1 Lista de Colunas do Índice PK_TIPO_MOVIMENTOS

Name	Code	Sort
FK_GRUPOS_MOVIMENTOS	FK_GRUPOS_MOVIMENTOS	Ascending
PK_TIPO_MOVIMENTOS	PK_TIPO_MOVIMENTOS	Ascending

1.3.17.8 Lista de Triggers da Tabela TIPO_MOVIMENTOS

Name	Code
------	------

Name	Code
TBIG_TIPO_MOVIMENTOS	TBIG_TIPO_MOVIMENTOS
TBU0_TIPO_MOVIMENTOS	TBU0_TIPO_MOVIMENTOS

1.3.18 Tabela TIPO_PAGAMENTOS

1.3.18.1 Card of table TIPO_PAGAMENTOS

Name	TIPO_PAGAMENTOS
Code	TIPO_PAGAMENTOS
Comment	Tabela que armazena os prazos defaults para vendas e/ou compras
Primary Key	PK_TIPO_PAGAMENTOS

1.3.18.2 Lista de Colunas da Tabela TIPO_PAGAMENTOS

Name	Data Type	Domain	Comment
PK_TIPO_PA GAMENTOS	SMALLINT	VALORS	Código do Tipo de Pagamento
FK_GRUPO S_MOVIMEN TOS	SMALLINT	VALORS	
FK_TIPO_D ESCONTOS	SMALLINT	VALORS	Código do tipo de desconto para esta tabela
DSC_TPG	VARCHAR(50)	DESCRICAO_50RQ	Descrição do Tipo de Pagamento
LST_PRZ	VARCHAR(50)	DESCRICAO_50	Lista de Prazos para parcelamentos definidos pelo usuário
FLAG_SEN	SMALLINT	FLAG_SIM_NAO	Flag que indica se esta condição de pagamento vai exigir uma senha para liberação da venda
FLAG_TINT	SMALLINT	FLAG_PARCELA	Tipo do Intervalo usado nesta Tabela 0 ==> Dias 1 ==> Semanas 2 ==> Meses 3 ==> Anos
FLAG_TVDA	SMALLINT	FLAG_TIPO_VENDA	Flag que indica o tipo da condição de pagamentos 0 ==> a Prazo 1 ==> a Vista 2 ==> Em Carteira (Acerto Futuro) 3 ==> Permuta
FLAG_ALT_ LST	SMALLINT	FLAG_SIM_NAO	Flag que indica se a lista de prazos pode ser alterada pelo usuário.
FLAG_DATA_ _BASE	SMALLINT	FLAG_DATA_BASE	Flag que indica qual a data base utilizada na condição de pagamentos 0 ==> A partir da Data de Faturamento 1 ==> A partir da Data do Pedido 2 ==> A partir da Data de Embarque 3 ==> A partir da Data Informada

Name	Data Type	Domain	Comment
FLAG_DEF_USER	SMALLINT	FLAG_SIM_NAO	Flag que indica que os prazos serão definidos pelo usuário
QTD_PAR	SMALLINT	VALORS	Quantidade de Parcelas da Condição de Pagamentos
INTRV_TAB	SMALLINT	VALORS	Intervalo conforme o Flag de periodicidade. Se periodicidade for dias e o intervalo for 30 ele calcula as parcelas de 30 em 30 dias.....
IND_TPGT	NUMERIC(9,4)	NUMERO_INDICE	índice que á aplicado para calcular o valor das parcelas.....
PER_JUR	NUMERIC(5,2)	NUMERO_PERCENTUAL	Percentua de juros sobre as parcelas
DSCT_TPG	NUMERIC(9,4)	NUMERO_INDICE	Desconto aplicado a esta condição de Pagamentos
IND_ENTR	NUMERIC(9,4)	NUMERO_INDICE	Índice utilizado para aplicar o valor da entrada
QTD_ENTR	SMALLINT	VALORS	Indica a quantidade de Parcelas utilizadas para calcular a entrada....
PER_ENTR	NUMERIC(5,2)	NUMERO_PERCENTUAL	Percentual da Entrada
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Operador que Incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.18.3 Dependências da Tabela TIPO_PAGAMENTOS

Name	Code	Class Name
FK_TIPOS_FRETES_TIPO_PAGAMENT	FK_TIPOS_FRETES_TIPO_PAGAMENT	Reference
FK_PEDIDOS_TIPO_PAGAMENT	FK_PEDIDOS_TIPO_PAGAMENT	Reference
FK_VINCULO_TPGTO_TIPO_PAGAMENT	FK_VINCULO_TPGTO_TIPO_PAGAMENT	Reference
FK_TIPO_PAGAMENTOS_CLIENTES	FK_TIPO_PAGAMENTOS_CLIENTES	Reference
FK_ORDENS_SERVIC_TIPO_PAGAMENT	FK_ORDENS_SERVIC_TIPO_PAGAMENT	Reference
FK_TIPO_PAGAMENT_TIPO_DESCONTO	FK_TIPO_PAGAMENT_TIPO_DESCONTO	Reference
FK_TIPO_PAGAMENT_GRUPOS_MOVIME	FK_TIPO_PAGAMENT_GRUPOS_MOVIME	Reference
TIPO_PAGAMENTOS	TIPO_PAGAMENTOS	Shortcut of table

Name	Code	Class Name
TIPO_PAGAMENTOS	TIPO_PAGAMENTOS	Shortcut of table
TIPO_PAGAMENTOS	TIPO_PAGAMENTOS	Shortcut of table

1.3.18.4 Lista de Chaves da Tabela TIPO_PAGAMENTOS

Name	Code	Primary
PK_TIPO_PAGAMENTOS	PK_TIPO_PAGAMENTOS	X

1.3.18.5 Chave PK_TIPO_PAGAMENTOS da Tabela TIPO_PAGAMENTOS

1.3.18.5.1 Lista de Colunas da Chave PK_TIPO_PAGAMENTOS

Name	Code
PK_TIPO_PAGAMENTOS	PK_TIPO_PAGAMENTOS

1.3.18.5.2 Dependências da Chave PK_TIPO_PAGAMENTOS

Name	Code	Class Name
PK_TIPO_PAGAMENTOS	PK_TIPO_PAGAMENTOS	Index
FK_TIPO_PAGAMENTOS_CLIENTES	FK_TIPO_PAGAMENTOS_CLIENTES	Reference
FK_TIPOS_FRETES_TIPO_PAGAMENTO	FK_TIPOS_FRETES_TIPO_PAGAMENTO	Reference
FK_PEDIDOS_TIPO_PAGAMENTO	FK_PEDIDOS_TIPO_PAGAMENTO	Reference
FK_VINCULO_TPGTO_TIPO_PAGAMENTO	FK_VINCULO_TPGTO_TIPO_PAGAMENTO	Reference
FK_ORDENS_SERVIC_TIPO_PAGAMENTO	FK_ORDENS_SERVIC_TIPO_PAGAMENTO	Reference

1.3.18.6 Lista de Índices TIPO_PAGAMENTOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_PAGAMENTOS	PK_TIPO_PAGAMENTOS	X		X		

1.3.18.7 Índice PK_TIPO_PAGAMENTOS da Tabela TIPO_PAGAMENTOS

1.3.18.7.1 Lista de Colunas do Índice PK_TIPO_PAGAMENTOS

Name	Code	Sort
PK_TIPO_PAGAMENTOS	PK_TIPO_PAGAMENTOS	Ascending

1.3.18.8 Lista de Triggers da Tabela TIPO_PAGAMENTOS

Name	Code
TBIG_TIPO_PAGAMENTOS	TBIG_TIPO_PAGAMENTOS

Name	Code
TBU0_TIPO_PAGAMENTOS	TBU0_TIPO_PAGAMENTOS

1.3.19 Tabela TIPO_PEDIDOS

1.3.19.1 Card of table TIPO_PEDIDOS

Name	TIPO_PEDIDOS
Code	TIPO_PEDIDOS
Comment	Tabela que armazena os tipos de pedidos que podem ser feitos
Primary Key	PK_TIPO_PEDIDOS

1.3.19.2 Lista de Colunas da Tabela TIPO_PEDIDOS

Name	Data Type	Domain	Comment
PK_TIPO_PEDIDOS	SMALLINT	VALORS	Código do tipo do desconto
FK_TIPO_DOCUMENTOS	SMALLINT	VALORS	Código do Tipo do Documento
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
DSC_TPED	VARCHAR(30)	DESCRICAO_30RQ	Descrição do tipo de pedido
FLAG_TPED	SMALLINT	FLAG_TIPO_PEDIDOS	Flag que indica o tipo de pedido 0 ==> Orçamentos 1 ==> Pedido Venda Representantes 2 ==> Pedido Venda Exportação 3 ==> Pedido Venda Filiais 4 ==> Pedido Venda Internet 5 ==> Pedido Compra 6 ==> Assistências
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	data e hora da inclusão do registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do operador da última alteração feita no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração feita no registro

1.3.19.3 Dependências da Tabela TIPO_PEDIDOS

Name	Code	Class Name
FK_PEDIDOS_TIPO_PEDIDOS	FK_PEDIDOS_TIPO_PEDIDOS	Reference
FK_PARAMETROS_PE_TIPO_PEDIDOS	FK_PARAMETROS_PE_TIPO_PEDIDOS	Reference
FK_TIPO_PEDIDOS_TIPO_DOCUMENT	FK_TIPO_PEDIDOS_TIPO_DOCUMENT	Reference
FK_TIPO_PEDIDOS_GRUPOS_MOVIME	FK_TIPO_PEDIDOS_GRUPOS_MOVIME	Reference

Name	Code	Primary
PK_TIPO_PEDIDOS	PK_TIPO_PEDIDOS	X

1.3.19.5 Chave PK_TIPO_PEDIDOS da Tabela TIPO_PEDIDOS

1.3.19.5.1 Lista de Colunas da Chave PK_TIPO_PEDIDOS

Name	Code
PK_TIPO_PEDIDOS	PK_TIPO_PEDIDOS

1.3.19.5.2 Dependências da Chave PK_TIPO_PEDIDOS

Name	Code	Class Name
PK_TIPO_PEDIDOS	PK_TIPO_PEDIDOS	Index
FK_PEDIDOS_TIPO_PEDIDOS	FK_PEDIDOS_TIPO_PEDIDOS	Reference
FK_PARAMETROS_PE_TIPO_PEDIDOS	FK_PARAMETROS_PE_TIPO_PEDIDOS	Reference

1.3.19.6 Lista de Índices TIPO_PEDIDOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_PEDIDOS	PK_TIPO_PEDIDOS	X		X		

1.3.19.7 Índice PK_TIPO_PEDIDOS da Tabela TIPO_PEDIDOS

1.3.19.7.1 Lista de Colunas do Índice PK_TIPO_PEDIDOS

Name	Code	Sort
PK_TIPO_PEDIDOS	PK_TIPO_PEDIDOS	Ascending

1.3.19.8 Lista de Triggers da Tabela TIPO_PEDIDOS

Name	Code
TBIG_TIPO_PEDIDOS	TBIG_TIPO_PEDIDOS
TBU0_TIPO_PEDIDOS	TBU0_TIPO_PEDIDOS

1.3.20 Tabela TIPO_PRAZO_ENTREGA

1.3.20.1 Card of table TIPO_PRAZO_ENTREGA

Name	TIPO_PRAZO_ENTREGA
Code	TIPO_PRAZO_ENTREGA
Comment	Tabela que informa os tipos de prazos de entrega do pedido
Primary Key	PK_TIPO_PRAZO_ENTREGA

1.3.20.2 Lista de Colunas da Tabela TIPO_PRAZO_ENTREGA

Name	Data Type	Domain	Comment
PK_TIPO_PRAZO_ENTREGA	SMALLINT	VALORS	Código
DSC_PRZE	VARCHAR(30)	DESCRICAO_30RQ	Descrição
QTD_PRV	SMALLINT	VALORS	Quantidade de dias prevista para a entrega. Somar esta quantidade à data do pedido
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.20.3 Dependências da Tabela TIPO_PRAZO_ENTREGA

Name	Code	Class Name
FK_PEDIDOS_TIPO_PRAZO_EN	FK_PEDIDOS_TIPO_PRAZO_EN	Reference
FK_PARAMETROS_PE_TIPO_PRAZO_EN	FK_PARAMETROS_PE_TIPO_PRAZO_EN	Reference
FK_TIPO_PRAZO_ENTREGA_CLIENTES	FK_TIPO_PRAZO_ENTREGA_CLIENTES	Reference
TIPO_PRAZO_ENTREGA	TIPO_PRAZO_ENTREGA	Shortcut of table

1.3.20.4 Lista de Chaves da Tabela TIPO_PRAZO_ENTREGA

Name	Code	Primary
PK_TIPO_PRAZO_ENTREGA	PK_TIPO_PRAZO_ENTREGA	X

1.3.20.5 Chave PK_TIPO_PRAZO_ENTREGA da Tabela TIPO_PRAZO_ENTREGA

1.3.20.5.1 Lista de Colunas da Chave PK_TIPO_PRAZO_ENTREGA

Name	Code
PK_TIPO_PRAZO_ENTREGA	PK_TIPO_PRAZO_ENTREGA

Name	Code	Class Name
PK_TIPO_PRAZO_ENTREGA	PK_TIPO_PRAZO_ENTREGA	Index
FK_TIPO_PRAZO_ENTREGA_CLIENTES	FK_TIPO_PRAZO_ENTREGA_CLIENTES	Reference
FK_PEDIDOS_TIPO_PRAZO_EN	FK_PEDIDOS_TIPO_PRAZO_EN	Reference
FK_PARAMETROS_PE_TIPO_PRAZO_EN	FK_PARAMETROS_PE_TIPO_PRAZO_EN	Reference

1.3.20.6 Lista de Índices TIPO_PRAZO_ENTREGA

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_PRAZO_ENTREGA	PK_TIPO_PRAZO_ENTREGA	X		X		

1.3.20.7 Índice PK_TIPO_PRAZO_ENTREGA da Tabela TIPO_PRAZO_ENTREGA

1.3.20.7.1 Lista de Colunas do Índice PK_TIPO_PRAZO_ENTREGA

Name	Code	Sort
PK_TIPO_PRAZO_ENTREGA	PK_TIPO_PRAZO_ENTREGA	Ascending

1.3.20.8 Lista de Triggers da Tabela TIPO_PRAZO_ENTREGA

Name	Code
TBU0_TIPO_PRAZO_ENTREGA	TBU0_TIPO_PRAZO_ENTREGA
TBIG_TIPO_PRAZO_ENTREGA	TBIG_TIPO_PRAZO_ENTREGA

1.3.21 Tabela TIPO_STATUS_PEDIDOS

1.3.21.1 Card of table TIPO_STATUS_PEDIDOS

Name	TIPO_STATUS_PEDIDOS
Code	TIPO_STATUS_PEDIDOS
Comment	Tabela que define o status do pedido
Primary Key	PK_TIPO_STATUS_PEDIDOS

1.3.21.2 Lista de Colunas da Tabela TIPO_STATUS_PEDIDOS

Name	Data Type	Domain	Comment
PK_TIPO_STATUS_PEDIDOS	SMALLINT	VALORS	Código do Status do Pedido
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
FK_TIPO_MOVIM_ESTQ	SMALLINT	VALORS	Código da tipo de movimentação de estoques
DSC_STT	VARCHAR(20)	DESCRICAO_20RQ	Descrição do Status
QTD_DAYS_NEXT	SMALLINT	VALORS	Quantidade de dias que indica o prazo máximo para troca do status do pedido
FLAG_OPEN	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido aberto
FLAG_RECB	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido recebido
FLAG_LIB	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido Liberado
FLAG_CANC	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido Cancelado
FLAG_PROD	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido em produção
FLAG_FAT	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido com faturamento parcial
FLAG_LIQ	SMALLINT	FLAG_SIM_NAO	Flag que indica um status de pedido liquidado (faturamento total)
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.21.3 Dependências da Tabela TIPO_STATUS_PEDIDOS

Name	Code	Class Name
FK_PEDIDOS_TIPO_STATUS_PEDIDOS	FK_PEDIDOS_TIPO_STATUS_PEDIDOS	Reference

Name	Code	Class Name
FK_PEDIDOS_STT_H_TIPO_STAT US_P	FK_PEDIDOS_STT_H_TIPO_STAT US_P	Reference
FK_PARAMETROS_PE_TIPO_STA TUS_P	FK_PARAMETROS_PE_TIPO_STA TUS_P	Reference
FK_TIPO_STATUS_P_TIPO_MOVI M_ES	FK_TIPO_STATUS_P_TIPO_MOVI M_ES	Reference
FK_TIPO_STATUS_GRUPOS_MOV IM	FK_TIPO_STATUS_GRUPOS_MOV IM	Reference

1.3.21.4 Lista de Chaves da Tabela TIPO_STATUS_PEDIDOS

Name	Code	Primary
PK_TIPO_STATUS_PEDIDOS	PK_TIPO_STATUS_PEDIDOS	X

1.3.21.5 Chave PK_TIPO_STATUS_PEDIDOS da Tabela TIPO_STATUS_PEDIDOS

1.3.21.5.1 Lista de Colunas da Chave PK_TIPO_STATUS_PEDIDOS

Name	Code
PK_TIPO_STATUS_PEDIDOS	PK_TIPO_STATUS_PEDIDOS

1.3.21.5.2 Dependências da Chave PK_TIPO_STATUS_PEDIDOS

Name	Code	Class Name
PK_TIPO_STATUS_PEDIDOS	PK_TIPO_STATUS_PEDIDOS	Index
FK_PEDIDOS_TIPO_STATUS_PED IDOS	FK_PEDIDOS_TIPO_STATUS_PED IDOS	Reference
FK_PEDIDOS_STT_H_TIPO_STAT US_P	FK_PEDIDOS_STT_H_TIPO_STAT US_P	Reference
FK_PARAMETROS_PE_TIPO_STA TUS_P	FK_PARAMETROS_PE_TIPO_STA TUS_P	Reference

1.3.21.6 Lista de Índices TIPO_STATUS_PEDIDOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_TIPO_STA TUS_PEDIDOS	PK_TIPO_STA TUS_PEDIDOS	X		X		

1.3.21.7 Índice PK_TIPO_STATUS_PEDIDOS da Tabela TIPO_STATUS_PEDIDOS

1.3.21.7.1 Lista de Colunas do Índice PK_TIPO_STATUS_PEDIDOS

Name	Code	Sort
PK_TIPO_STATUS_PEDIDOS	PK_TIPO_STATUS_PEDIDOS	Ascending

1.3.21.8 Lista de Triggers da Tabela TIPO_STATUS_PEDIDOS

Name	Code
------	------

Name	Code
TBIG_TIPO_STATUS_PEDIDOS	TBIG_TIPO_STATUS_PEDIDOS
TBU0_TIPO_STATUS_PEDIDOS	TBU0_TIPO_STATUS_PEDIDOS

1.3.22 Tabela VINCULO_TPGTO_FINALIZ

1.3.22.1 Card of table VINCULO_TPGTO_FINALIZ

Name	VINCULO_TPGTO_FINALIZ
Code	VINCULO_TPGTO_FINALIZ
Comment	Vincula os tipos de pagamento às finalizadoras
Primary Key	PK_VINCULO_TPGTO_FINALIZ

1.3.22.2 Lista de Colunas da Tabela VINCULO_TPGTO_FINALIZ

Name	Data Type	Domain	Comment
FK_FINALIZ ADORAS	SMALLINT	VALORS	Código
FK_TIPO_PA GAMENTOS	SMALLINT	VALORS	Código do Tipo de Pagamento
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do Operador que incluiu o registro
DTHR_INC	TIMESTAMP	DATA_HORA_DEF	Data e hora da inclusão no registro
OPE_ALT	VARCHAR(10)	NOME_OPERADOR	Nome do Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e hora da última alteração no registro

1.3.22.3 Dependências da Tabela VINCULO_TPGTO_FINALIZ

Name	Code	Class Name
FK_VINCULO_TPGTO_FINALIZAD ORAS	FK_VINCULO_TPGTO_FINALIZAD ORAS	Reference
FK_VINCULO_TPGTO_TIPO_PAG AMENT	FK_VINCULO_TPGTO_TIPO_PAG AMENT	Reference

1.3.22.4 Lista de Chaves da Tabela VINCULO_TPGTO_FINALIZ

Name	Code	Primary
PK_VINCULO_TPGTO_FINALIZ	PK_VINCULO_TPGTO_FINALIZ	X

1.3.22.5 Chave PK_VINCULO_TPGTO_FINALIZ da Tabela VINCULO_TPGTO_FINALIZ

1.3.22.5.1 Lista de Colunas da Chave PK_VINCULO_TPGTO_FINALIZ

Name	Code
FK_FINALIZADORAS	FK_FINALIZADORAS
FK_TIPO_PAGAMENTOS	FK_TIPO_PAGAMENTOS

1.3.22.5.2 Dependências da Chave PK_VINCULO_TPGTO_FINALIZ

Name	Code	Class Name
PK_VINCULO_TPGTO_FINALIZ	PK_VINCULO_TPGTO_FINALIZ	Index

1.3.22.6 Lista de Índices VINCULO_TPGTO_FINALIZ

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_VINCULO_TPGTO_FINALIZ	PK_VINCULO_TPGTO_FINALIZ	X		X		

1.3.22.7 Índice PK_VINCULO_TPGTO_FINALIZ da Tabela VINCULO_TPGTO_FINALIZ

1.3.22.7.1 Lista de Colunas do Índice PK_VINCULO_TPGTO_FINALIZ

Name	Code	Sort
FK_FINALIZADORAS	FK_FINALIZADORAS	Ascending
FK_TIPO_PAGAMENTOS	FK_TIPO_PAGAMENTOS	Ascending

1.3.22.8 Lista de Triggers da Tabela VINCULO_TPGTO_FINALIZ

Name	Code
TBU0_VINCULO_TPGTO_FINALIZ	TBU0_VINCULO_TPGTO_FINALIZ

1.4 Stored Procedures**1.4.1 Procedure STP_GET_CONF_COMPO_TREF****1.4.1.1 Card of procedure STP_GET_CONF_COMPO_TREF**

<i>Name</i>	STP_GET_CONF_COMPO_TREF
<i>Code</i>	STP_GET_CONF_COMPO_TREF
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.1.2 Code preview of procedure STP_GET_CONF_COMPO_TREF

set term ^;

```

create procedure STP_GET_CONF_COMPO_TREF (
  P_FK_EMPRESAS      integer,
  P_FK_TABELA_PRECOS  smallint,
  P_FK_PRODUTOS      integer,
  P_FK_COMPONENTES    smallint,
  P_FK_TIPO_REFERENCIAS integer,
  P_FK_TIPO_ACABAMENTOS integer
)
returns (
  R_PRE_VDA          numeric(11, 4),
  R_QTD_PDR          numeric(09, 4),
  R_QTD_PERS         numeric(09, 4),
  R_PRE_UNT_PDR      numeric(09, 4),
  R_PRE_UNT_PERS     numeric(09, 4),
  R_TOT_INS_BXA      numeric(09, 4),
  R_TOT_INS_CLC      numeric(09, 4)
)
as
  declare variable PrecoUnitario double precision;
  declare variable FlagOutOfRange smallint;
begin
  select Acp.PRE_VDA, Aca.QTD_PDR, Aca.QTD_PERS
  from TIPO_ACABAMENTOS Tac, ACABAMENTOS Aca,
       TIPO_REFERENCIAS Tre, ACABAMENTO_PRECOS Acp
  where Acp.FK_EMPRESAS = :P_FK_EMPRESAS
  and Acp.fk_produtos = Aca.FK_PRODUTOS
  and Acp.FK_TIPO_ACABAMENTOS = Tre.FK_TIPO_ACABAMENTOS
  and Acp.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_ACABAMENTOS
  and Acp.FK_TIPO_REFERENCIAS = Tre.PK_TIPO_REFERENCIAS
  and Acp.FK_TIPO_REFERENCIAS = :P_FK_TIPO_REFERENCIAS
  and Acp.FK_TABELA_PRECOS = :P_FK_TABELA_PRECOS
  and Aca.FK_PRODUTOS = :P_FK_PRODUTOS
  and ((Aca.FK_COMPONENTES = :P_FK_COMPONENTES)
  or (0 = :P_FK_COMPONENTES))
  and Tac.PK_TIPO_ACABAMENTOS = Aca.FK_TIPO_ACABAMENTOS
  and Tre.FK_TIPO_ACABAMENTOS = Aca.FK_TIPO_ACABAMENTOS
  order by Tac.DSC_ACABM, Tre.DSC_REF
  into :R_PRE_VDA, :R_QTD_PDR, :R_QTD_PERS;
  if (R_PRE_VDA is null) then
    R_PRE_VDA = 0;
  if (R_QTD_PDR is null) then
    R_QTD_PDR = 0;

```

```

if (R_QTD_PERS is null) then
  R_QTD_PERS = 0;
FlagOutOfRange = 0;
select sum(Aca.QTD_PDR * Cmp.QTD_COMP),
       sum(Aca.QTD_PDR * Cmp.QTD_COMP)
  from ACABAMENTOS Aca, COMPONENTES Cmp
 where Cmp.FK_PRODUTOS      = :P_FK_PRODUTOS
    and Aca.FK_PRODUTOS      = Cmp.FK_PRODUTOS
    and Aca.FK_COMPONENTES    = Cmp.FK_TIPO_COMPONENTES
    and Aca.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_ACABAMENTOS
 into :R_TOT_INS_BXA, :R_TOT_INS_CLC;
if ((R_TOT_INS_BXA is null) or (R_TOT_INS_BXA = 0) or
    (R_TOT_INS_CLC is null) or (R_TOT_INS_CLC = 0)) then
begin
  R_TOT_INS_BXA = 1;
  R_TOT_INS_CLC = 1;
  FlagOutOfRange = -1;
end
PrecoUnitario = R_PRE_VDA / R_TOT_INS_CLC;
R_PRE_UNT_PERS = PrecoUnitario * R_QTD_PERS;
PrecoUnitario = R_PRE_VDA / R_TOT_INS_BXA;
R_PRE_UNT_PDR = PrecoUnitario * R_QTD_PDR;
if (FlagOutOfRange = -1) then
begin
  R_TOT_INS_BXA = 0;
  R_TOT_INS_CLC = 0;
end
suspend;
end^

set term ^;

```

grant EXECUTE on procedure STP_GET_CONF_COMPO_TREF to PUBLIC;

1.4.1.3 Text of procedure STP_GET_CONF_COMPO_TREF

```

set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS      integer,
  P_FK_TABELA_PRECOS  smallint,
  P_FK_PRODUTOS      integer,
  P_FK_COMPONENTES    smallint,
  P_FK_TIPO_REFERENCIAS integer,
  P_FK_TIPO_ACABAMENTOS integer
)
returns (
  R_PRE_VDA          numeric(11, 4),
  R_QTD_PDR          numeric(09, 4),
  R_QTD_PERS         numeric(09, 4),
  R_PRE_UNT_PDR      numeric(09, 4),
  R_PRE_UNT_PERS     numeric(09, 4),
  R_TOT_INS_BXA      numeric(09, 4),
  R_TOT_INS_CLC      numeric(09, 4)
)
as
  declare variable PrecoUnitario double precision;
  declare variable FlagOutOfRange smallint;

```



```

begin
select Acp.PRE_VDA, Aca.QTD_PDR, Aca.QTD_PERS
from TIPO_ACABAMENTOS Tac, ACABAMENTOS Aca,
     TIPO_REFERENCIAS Tre, ACABAMENTO_PRECOS Acp
where Acp.FK_EMPRESAS      = :P_FK_EMPRESAS
and Acp.fk_produtos      = Aca.FK_PRODUTOS
and Acp.FK_TIPO_ACABAMENTOS = Tre.FK_TIPO_ACABAMENTOS
and Acp.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_ACABAMENTOS
and Acp.FK_TIPO_REFERENCIAS = Tre.PK_TIPO_REFERENCIAS
and Acp.FK_TIPO_REFERENCIAS = :P_FK_TIPO_REFERENCIAS
and Acp.FK_TABELA_PRECOS   = :P_FK_TABELA_PRECOS
and Aca.FK_PRODUTOS       = :P_FK_PRODUTOS
and ((Aca.FK_COMPONENTES   = :P_FK_COMPONENTES)
or (0                       = :P_FK_COMPONENTES))
and Tac.PK_TIPO_ACABAMENTOS = Aca.FK_TIPO_ACABAMENTOS
and Tre.FK_TIPO_ACABAMENTOS = Aca.FK_TIPO_ACABAMENTOS
order by Tac.DSC_ACABM, Tre.DSC_REF
into :R_PRE_VDA, :R_QTD_PDR, :R_QTD_PERS;
if (R_PRE_VDA is null) then
  R_PRE_VDA = 0;
if (R_QTD_PDR is null) then
  R_QTD_PDR = 0;
if (R_QTD_PERS is null) then
  R_QTD_PERS = 0;
FlagOutOfRange = 0;
select sum(Aca.QTD_PDR * Cmp.QTD_COMP),
      sum(Aca.QTD_PDR * Cmp.QTD_COMP)
from ACABAMENTOS Aca, COMPONENTES Cmp
where Cmp.FK_PRODUTOS      = :P_FK_PRODUTOS
and Aca.FK_PRODUTOS       = Cmp.FK_PRODUTOS
and Aca.FK_COMPONENTES    = Cmp.FK_TIPO_COMPONENTES
and Aca.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_ACABAMENTOS
into :R_TOT_INS_BXA, :R_TOT_INS_CLC;
if ((R_TOT_INS_BXA is null) or (R_TOT_INS_BXA = 0) or
    (R_TOT_INS_CLC is null) or (R_TOT_INS_CLC = 0)) then
begin
  R_TOT_INS_BXA = 1;
  R_TOT_INS_CLC = 1;
  FlagOutOfRange = -1;
end
PrecoUnitario = R_PRE_VDA / R_TOT_INS_CLC;
R_PRE_UNT_PERS = PrecoUnitario * R_QTD_PERS;
PrecoUnitario = R_PRE_VDA / R_TOT_INS_BXA;
R_PRE_UNT_PDR = PrecoUnitario * R_QTD_PDR;
if (FlagOutOfRange = -1) then
begin
  R_TOT_INS_BXA = 0;
  R_TOT_INS_CLC = 0;
end
suspend;
end^

set term ;^

```

1.4.1.4 List of diagrams containing the procedure STP_GET_CONF_COMPO_TREF

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.1.5 List of permissions of the procedure STP_GET_CONF_COMPO_TREF

<i>User</i>	<i>Grant</i>	<i>Grant With Option</i>	<i>Revoke</i>	<i>Revoke With Cascade</i>
PUBLIC	EXECUTE			

1.4.2 Procedure STP_GET_PK_DESCONTOS**1.4.2.1 Card of procedure STP_GET_PK_DESCONTOS**

<i>Name</i>	STP_GET_PK_DESCONTOS
<i>Code</i>	STP_GET_PK_DESCONTOS
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.2.2 Code preview of procedure STP_GET_PK_DESCONTOS

set term ^;

```

create procedure STP_GET_PK_DESCONTOS (
  P_FK_TIPO_DESCONTOS integer
)
returns (
  R_PK_DESCONTOS smallint
)
as
begin
  R_PK_DESCONTOS = -1;
  if ((P_FK_TIPO_DESCONTOS is not null) and (P_FK_TIPO_DESCONTOS > 0)) then
  begin
    select KC_DESCONTOS
    from TIPO_DESCONTOS
    where PK_TIPO_DESCONTOS = :P_FK_TIPO_DESCONTOS
    into :R_PK_DESCONTOS;
    if ((R_PK_DESCONTOS is not null) and (R_PK_DESCONTOS >= 0)) then
    begin
      R_PK_DESCONTOS = R_PK_DESCONTOS + 1;
      update TIPO_DESCONTOS set
        KC_DESCONTOS = :R_PK_DESCONTOS
        where PK_TIPO_DESCONTOS = :P_FK_TIPO_DESCONTOS;
    end
  end
  suspend;
end ^

```

set term ^;

grant execute on procedure STP_GET_PK_DESCONTOS to public;

1.4.2.3 Text of procedure STP_GET_PK_DESCONTOS

set term ^;

```

create procedure %PROC% (
  P_FK_TIPO_DESCONTOS integer
)
returns (

```

```
R_PK_DESCONTOS smallint
)
as
begin
  R_PK_DESCONTOS = -1;
  if ((P_FK_TIPO_DESCONTOS is not null) and (P_FK_TIPO_DESCONTOS > 0)) then
  begin
    select KC_DESCONTOS
    from TIPO_DESCONTOS
    where PK_TIPO_DESCONTOS = :P_FK_TIPO_DESCONTOS
    into :R_PK_DESCONTOS;
    if ((R_PK_DESCONTOS is not null) and (R_PK_DESCONTOS >= 0)) then
    begin
      R_PK_DESCONTOS = R_PK_DESCONTOS + 1;
      update TIPO_DESCONTOS set
        KC_DESCONTOS = :R_PK_DESCONTOS
        where PK_TIPO_DESCONTOS = :P_FK_TIPO_DESCONTOS;
    end
  end
  suspend;
end ^

set term ;^
```

1.4.2.4 List of diagrams containing the procedure STP_GET_PK_DESCONTOS

Name	Code
SysPed	SYSPED

1.4.3 Procedure STP_GET_PK_ORDER

1.4.3.1 Card of procedure STP_GET_PK_ORDER

Name	STP_GET_PK_ORDER
Code	STP_GET_PK_ORDER
Parent	Package 'SysPed'
Type	Procedure

1.4.3.2 Code preview of procedure STP_GET_PK_ORDER

```
set term ^;

create procedure STP_GET_PK_ORDER (
  P_FK_EMPRESAS integer,
  P_FK_TIPO_PEDIDOS smallint,
  P_FLAG_ORDER_TMP smallint
)
returns (
  R_PK_ORDER integer
)
as
  declare variable TipoDoc integer;
begin
  R_PK_ORDER = null;
  if (P_FLAG_ORDER_TMP is null) then
    P_FLAG_ORDER_TMP = 0;
  if ((P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
    (P_FLAG_ORDER_TMP = 0)) then
```

```
begin
select FK_TIPO_DOCUMENTOS
from TIPO_PEDIDOS
where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
into :TipoDoc;
if ((TipoDoc is not null) and (TipoDoc > 0)) then
begin
select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :TipoDoc
into :R_PK_ORDER;
if (R_PK_ORDER is null) then
begin
insert into TIPO_DOCUMENTOS_NUMERACAO
(FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)
values
(:P_FK_EMPRESAS, :TipoDoc, 0);
R_PK_ORDER = 0;
end
R_PK_ORDER = R_PK_ORDER + 1;
update TIPO_DOCUMENTOS_NUMERACAO set
NUM_DOC = :R_PK_ORDER
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :TipoDoc;
end
end
if (P_FLAG_ORDER_TMP > 0) then
P_FLAG_ORDER_TMP = Gen_Id(PEDIDOS_TMP, 1);
suspend;
end^

set term ;^

grant execute on procedure STP_GET_PK_ORDER to public;
```

1.4.3.3 Text of procedure STP_GET_PK_ORDER

```
set term ^;

create procedure %PROC% (
P_FK_EMPRESAS integer,
P_FK_TIPO_PEDIDOS smallint,
P_FLAG_ORDER_TMP smallint
)
returns (
R_PK_ORDER integer
)
as
declare variable TipoDoc integer;
begin
R_PK_ORDER = null;
if (P_FLAG_ORDER_TMP is null) then
P_FLAG_ORDER_TMP = 0;
if ((P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
(P_FLAG_ORDER_TMP = 0)) then
begin
select FK_TIPO_DOCUMENTOS
from TIPO_PEDIDOS
```

```
where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
into :TipoDoc;
if ((TipoDoc is not null) and (TipoDoc > 0)) then
begin
select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :TipoDoc
into :R_PK_ORDER;
if (R_PK_ORDER is null) then
begin
insert into TIPO_DOCUMENTOS_NUMERACAO
(FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)
values
(:P_FK_EMPRESAS, :TipoDoc, 0);
R_PK_ORDER = 0;
end
R_PK_ORDER = R_PK_ORDER + 1;
update TIPO_DOCUMENTOS_NUMERACAO set
NUM_DOC = :R_PK_ORDER
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :TipoDoc;
end
end
if (P_FLAG_ORDER_TMP > 0) then
P_FLAG_ORDER_TMP = Gen_Id(PEDIDOS_TMP, 1);
suspend;
end^
```

set term ;^

grant execute on procedure %PROC% to public;

1.4.3.4 List of diagrams containing the procedure STP_GET_PK_ORDER

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.4 Procedure STP_GET_PK_PEDIDOS_DESCONTOS

1.4.4.1 Card of procedure STP_GET_PK_PEDIDOS_DESCONTOS

<i>Name</i>	STP_GET_PK_PEDIDOS_DESCONTOS
<i>Code</i>	STP_GET_PK_PEDIDOS_DESCONTOS
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.4.2 Code preview of procedure STP_GET_PK_PEDIDOS_DESCONTOS

set term ^;

```
create procedure STP_GET_PK_PEDIDOS_DESCONTOS (
P_FK_EMPRESAS integer,
P_FK_TIPO_PEDIDOS integer,
P_FK_PEDIDOS integer
)
returns (
R_PK_PEDIDOS_DESCONTOS smallint
```

```

)
as
begin
  R_PK_PEDIDOS_DESCONTOS = -1;
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
      (P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
      (P_FK_PEDIDOS is not null) and (P_FK_PEDIDOS > 0)) then
    begin
      select KC_PEDIDOS_DESCONTOS
      from PEDIDOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
            and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
            and PK_PEDIDOS = :P_FK_PEDIDOS
      into :R_PK_PEDIDOS_DESCONTOS;
    end if ((R_PK_PEDIDOS_DESCONTOS is not null) and
            (R_PK_PEDIDOS_DESCONTOS >= 0)) then
    begin
      R_PK_PEDIDOS_DESCONTOS = R_PK_PEDIDOS_DESCONTOS + 1;
      update PEDIDOS set
        KC_PEDIDOS_DESCONTOS = :R_PK_PEDIDOS_DESCONTOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
            and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
            and PK_PEDIDOS = :P_FK_PEDIDOS;
    end
  end
suspend;
end ^

```

set term ^;

grant execute on procedure STP_GET_PK_PEDIDOS_DESCONTOS to public;

1.4.4.3 Text of procedure STP_GET_PK_PEDIDOS_DESCONTOS

set term ^;

```

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_FK_TIPO_PEDIDOS integer,
  P_FK_PEDIDOS integer
)
returns (
  R_PK_PEDIDOS_DESCONTOS smallint
)
as
begin
  R_PK_PEDIDOS_DESCONTOS = -1;
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
      (P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
      (P_FK_PEDIDOS is not null) and (P_FK_PEDIDOS > 0)) then
    begin
      select KC_PEDIDOS_DESCONTOS
      from PEDIDOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
            and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
            and PK_PEDIDOS = :P_FK_PEDIDOS
      into :R_PK_PEDIDOS_DESCONTOS;
    end if ((R_PK_PEDIDOS_DESCONTOS is not null) and
            (R_PK_PEDIDOS_DESCONTOS >= 0)) then
    begin
      R_PK_PEDIDOS_DESCONTOS = R_PK_PEDIDOS_DESCONTOS + 1;
      update PEDIDOS set
        KC_PEDIDOS_DESCONTOS = :R_PK_PEDIDOS_DESCONTOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
            and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
            and PK_PEDIDOS = :P_FK_PEDIDOS;
    end
  end
suspend;
end ^

```

```
begin
  R_PK_PEDIDOS_DESCONTOS = R_PK_PEDIDOS_DESCONTOS + 1;
update PEDIDOS set
  KC_PEDIDOS_DESCONTOS = :R_PK_PEDIDOS_DESCONTOS
where FK_EMPRESAS    = :P_FK_EMPRESAS
  and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
  and PK_PEDIDOS      = :P_FK_PEDIDOS;
end
end
suspend;
end ^

set term ;^
```

1.4.4.4 List of diagrams containing the procedure STP_GET_PK_PEDIDOS_DESCONTOS

Name	Code
SysPed	SYSPED

1.4.5 Procedure STP_GET_PK_PEDIDOS_HISTORICOS

1.4.5.1 Card of procedure STP_GET_PK_PEDIDOS_HISTORICOS

Name	STP_GET_PK_PEDIDOS_HISTORICOS
Code	STP_GET_PK_PEDIDOS_HISTORICOS
Parent	Package 'SysPed'
Type	Procedure

1.4.5.2 Code preview of procedure STP_GET_PK_PEDIDOS_HISTORICOS

```
set term ^;

create procedure STP_GET_PK_PEDIDOS_HISTORICOS (
  P_FK_EMPRESAS    integer,
  P_FK_TIPO_PEDIDOS integer,
  P_FK_PEDIDOS      integer
)
returns (
  R_PK_PEDIDOS_HISTORICOS smallint
)
as
begin
  R_PK_PEDIDOS_HISTORICOS = -1;
if ((P_FK_EMPRESAS    is not null) and (P_FK_EMPRESAS    > 0) and
  ( P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
  ( P_FK_PEDIDOS      is not null) and (P_FK_PEDIDOS      > 0)) then
begin
  select KC_PEDIDOS_HISTORICOS
  from PEDIDOS
  where FK_EMPRESAS    = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and PK_PEDIDOS      = :P_FK_PEDIDOS
  into :R_PK_PEDIDOS_HISTORICOS;
if ((R_PK_PEDIDOS_HISTORICOS is not null) and
  ( R_PK_PEDIDOS_HISTORICOS >= 0)) then
begin
  R_PK_PEDIDOS_HISTORICOS = R_PK_PEDIDOS_HISTORICOS + 1;
  update PEDIDOS set
```

```
KC_PEDIDOS_HISTORICOS = :R_PK_PEDIDOS_HISTORICOS
where FK_EMPRESAS    = :P_FK_EMPRESAS
and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
and PK_PEDIDOS      = :P_FK_PEDIDOS;
end
end
suspend;
end ^

set term ^;
```

grant execute on procedure STP_GET_PK_PEDIDOS_HISTORICOS to public;

1.4.5.3 Text of procedure STP_GET_PK_PEDIDOS_HISTORICOS

```
set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS    integer,
  P_FK_TIPO_PEDIDOS integer,
  P_FK_PEDIDOS     integer
)
returns (
  R_PK_PEDIDOS_HISTORICOS smallint
)
as
begin
  R_PK_PEDIDOS_HISTORICOS = -1;
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
    ( P_FK_TIPO_PEDIDOS is not null) and (P_FK_TIPO_PEDIDOS > 0) and
    ( P_FK_PEDIDOS is not null) and (P_FK_PEDIDOS > 0)) then
  begin
    select KC_PEDIDOS_HISTORICOS
    from PEDIDOS
    where FK_EMPRESAS    = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and PK_PEDIDOS      = :P_FK_PEDIDOS
    into :R_PK_PEDIDOS_HISTORICOS;
    if ((R_PK_PEDIDOS_HISTORICOS is not null) and
      ( R_PK_PEDIDOS_HISTORICOS >= 0)) then
    begin
      R_PK_PEDIDOS_HISTORICOS = R_PK_PEDIDOS_HISTORICOS + 1;
      update PEDIDOS set
        KC_PEDIDOS_HISTORICOS = :R_PK_PEDIDOS_HISTORICOS
      where FK_EMPRESAS    = :P_FK_EMPRESAS
      and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
      and PK_PEDIDOS      = :P_FK_PEDIDOS;
    end
  end
  suspend;
end ^

set term ^;
```

1.4.5.4 List of diagrams containing the procedure STP_GET_PK_PEDIDOS_HISTORICOS

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.6 Procedure STP_GET_PK_PRODUCT**1.4.6.1 Card of procedure STP_GET_PK_PRODUCT**

<i>Name</i>	STP_GET_PK_PRODUCT
<i>Code</i>	STP_GET_PK_PRODUCT
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.6.2 Code preview of procedure STP_GET_PK_PRODUCT

set term ^;

```
create procedure STP_GET_PK_PRODUCT (
  P_FK_PRODUTO      varchar(30),
  P_CODE_TYPE       smallint
)
returns (
  R_PK_PRODUCT      integer
)
as
begin
/* 0 ==> Referencia do Produto */
/* 1 ==> Codigo do Produto    */
/* 2 ==> Codigo de Barras     */
/* 3 ==> Codigo do Fornecedor */
  R_PK_PRODUCT = 0;
  if (P_CODE_TYPE = 1) then
  begin
    R_PK_PRODUCT = cast(P_FK_PRODUTO as Integer);
    suspend;
  end
  begin
    select FK_PRODUTOS from PRODUTOS_CODIGOS
    where COD_REF  = :P_FK_PRODUTO
    and FLAG_TCODE = :P_CODE_TYPE
    into :R_PK_PRODUCT;
    suspend;
  end
end ^
```

set term ^;

grant execute on procedure STP_GET_PK_PRODUCT to public;

1.4.6.3 Text of procedure STP_GET_PK_PRODUCT

set term ^;

```
create procedure %PROC% (
  P_FK_PRODUTO      varchar(30),
  P_CODE_TYPE       smallint
)
returns (
  R_PK_PRODUCT      integer
)
as
begin
```

```

/* 0 ==> Referencia do Produto */
/* 1 ==> Codigo do Produto    */
/* 2 ==> Codigo de Barras     */
/* 3 ==> Codigo do Fornecedor */
R_PK_PRODUCT = 0;
if (P_CODE_TYPE = 1) then
begin
  R_PK_PRODUCT = cast(P_FK_PRODUTO as Integer);
  suspend;
end
begin
  select FK_PRODUTOS from PRODUTOS_CODIGOS
  where COD_REF  = :P_FK_PRODUTO
    and FLAG_TCODE = :P_CODE_TYPE
    into :R_PK_PRODUCT;
  suspend;
end
end ^

set term ;^

```

1.4.6.4 List of diagrams containing the procedure STP_GET_PK_PRODUCT

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.7 Procedure STP_GET_PRODUCT_ATAxes

1.4.7.1 Card of procedure STP_GET_PRODUCT_ATAxes

<i>Name</i>	STP_GET_PRODUCT_ATAxes
<i>Code</i>	STP_GET_PRODUCT_ATAxes
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.7.2 Code preview of procedure STP_GET_PRODUCT_ATAxes

set term ^;

```

create procedure STP_GET_PRODUCT_ATAxes (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer,
  P_FLAG_ES      smallint,
  P_FLAG_CNS     smallint,
  P_FK_PAIS_ORGM smallint,
  P_FK_PAIS_DSTN smallint,
  P_FK_ESTD_ORGM varchar(2),
  P_FK_ESTD_DSTN varchar(2),
  P_FK_MUNICIPIOS integer,
  P_TYPE_PRINTER integer
)
returns (
  R_RED_BASC      numeric(5, 2),
  R_ALQ_ICMS      numeric(5, 2),
  R_ALQ_ICMSS     numeric(5, 2),
  R_ALQ_IPI       numeric(5, 2),
  R_ALQ_OTR       numeric(5, 2),

```

```

R_COD_ICMS_ECF smallint,
R_COD_ICMSS_ECF smallint,
R_COD_IPI_ECF smallint,
R_COD_ISS_ECF smallint,
R_COD_OTR_ECF smallint,
R_FLAG_SRV smallint,
R_STATUS smallint
)
as
declare variable FkTipoImpostos smallint;
declare variable FlagImpst smallint;
declare variable PaisDestino smallint;
declare variable PaisOrigem smallint;
declare variable EstDestino varchar(2);
declare variable EstOrigem varchar(2);
declare variable AlqtSaiCons numeric(5, 2);
declare variable AlqtSai numeric(5, 2);
declare variable ArbtIcmss numeric(5, 2);
declare variable CodAlqtSaiCons smallint;
declare variable CodAlqtSai smallint;
begin
R_RED_BASC = 0.0;
R_ALQ_ICMS = 0.0;
R_ALQ_ICMSS = 0.0;
R_ALQ_IPI = 0.0;
R_ALQ_OTR = 0.0;
R_COD_ICMS_ECF = 0;
R_COD_ICMSS_ECF = 0;
R_COD_IPI_ECF = 0;
R_COD_ISS_ECF = 0;
R_COD_OTR_ECF = 0;
R_FLAG_SRV = 0;
R_STATUS = 0;
/* select fisco info
0 ==> Base para ICMS
1 ==> Base para ICMS Substituicao
2 ==> Base para ICMS Isento
3 ==> Base para IPI
4 ==> Base para ISS
5 ==> Base para Outros
*/
PaisOrigem = P_FK_PAIS_ORGM;
PaisDestino = P_FK_PAIS_DSTN;
EstOrigem = P_FK_ESTD_ORGM;
EstDestino = P_FK_ESTD_DSTN;
for select Pri.FK_TIPO_IMPOSTOS, Tim.FLAG_IMPST, Tim.RED_BASC
from PRODUTOS_IMPOSTOS Pri, TIPO_IMPOSTOS Tim
where Pri.FK_EMPRESAS = :P_FK_EMPRESAS
and Pri.FK_PRODUTOS = :P_FK_PRODUTOS
and Tim.PK_TIPO_IMPOSTOS = Pri.FK_TIPO_IMPOSTOS
into :FkTipoImpostos, :FlagImpst, :R_RED_BASC do
begin
if (P_FLAG_ES = 0) then
select FK_PAISES, FK_ESTADOS /* select country and state of the company */
from EMPRESAS
where PK_EMPRESAS = :P_FK_EMPRESAS
into :PaisDestino, :EstDestino;
else

```

```
select FK_PAISES, FK_ESTADOS /* select country and state of the company */
from EMPRESAS
where PK_EMPRESAS = :P_FK_EMPRESAS
into :PaisOrigem, :EstOrigem;
select ALQT_CNSF, ALQT_IMPST, ALQT_ARBT
from ALIQUOTAS
where FK_PAISES__ORGM = :PaisOrigem
and FK_ESTADOS__ORGM = :EstOrigem
and FK_PAISES__DST = :PaisDestino
and FK_ESTADOS__DST = :EstDestino
and FK_TIPO_IMPOSTOS = :FkTipoImpostos
into :AlqtSaiCons, :AlqtSai, :ArbtIcmss;
if (((AlqtSai is null) or (AlqtSai <= 0)) and
(FlagImpst not in (1, 2))) then
R_STATUS = 30; /* Tax of product not found */
if (FlagImpst is null) then
R_STATUS = 40; /* whithout fiscal classification */
R_ALQ_ICMS = 0;
R_ALQ_ICMSS = 0;
R_ALQ_IPI = 0;
R_ALQ_OTR = 0;
if (FlagImpst = 0) then
if (P_FLAG_CNS = 1) then
R_ALQ_ICMS = AlqtSaiCons;
else
R_ALQ_ICMS = AlqtSai;
if (FlagImpst = 1) then
if (P_FLAG_CNS = 1) then
R_ALQ_ICMSS = AlqtSaiCons;
else
R_ALQ_ICMSS = AlqtSai;
if (FlagImpst = 3) then
R_ALQ_IPI = AlqtSai;
if (FlagImpst = 4) then
R_FLAG_SRV = 1;
if (FlagImpst = 5) then
R_ALQ_OTR = AlqtSai;
if ((P_FLAG_CNS = 1) and (P_TYPE_PRINTER is not null) and
(P_TYPE_PRINTER > 0)) then
begin
select COD_ALQT_CNSF, COD_ALQT
from ALIQUOTAS_IMP_FISCAL
where FK_PAISES__ORGM = :PaisOrigem
and FK_ESTADOS__ORGM = :EstOrigem
and FK_PAISES__DST = :PaisDestino
and FK_ESTADOS__DST = :PaisDestino
and FK_TIPO_IMPOSTOS = :FkTipoImpostos
and PK_ALIQUOTAS_IMP_FISCAL = :P_TYPE_PRINTER
into :CodAlqtSaiCons, :CodAlqtSai;
R_COD_ICMS_ECF = 0;
R_COD_ICMSS_ECF = 0;
R_COD_IPI_ECF = 0;
R_COD_ISS_ECF = 0;
R_COD_OTR_ECF = 0;
if (FlagImpst = 0) then
if (P_FLAG_CNS = 1) then
R_COD_ICMS_ECF = CodAlqtSaiCons;
else
```

```
R_COD_ICMS_ECF = CodAlqtSai;
if (FlagImpst = 1) then
  if (P_FLAG_CNS = 1) then
    R_COD_ICMSS_ECF = CodAlqtSaiCons;
  else
    R_COD_ICMSS_ECF = CodAlqtSai;
  if (FlagImpst = 3) then
    R_COD_IPI_ECF = CodAlqtSai;
  if (FlagImpst = 4) then
    begin
      select COD_ISS_ECF
      from MUNICIPIOS_ALIQUOTAS
      where FK_PAISES          = :PaisOrigem
      and FK_ESTADOS          = :EstOrigem
      and FK_MUNICIPIOS       = :P_FK_MUNICIPIOS
      and PK_MUNICIPIOS_ALIQUOTAS = :P_TYPE_PRINTER
      into :R_COD_ISS_ECF;
    end /* end of if */
  if (FlagImpst = 5) then
    R_COD_OTR_ECF = CodAlqtSai;
  end /* end of if */
end
suspend;
end ^

set term ;^

grant execute on procedure STP_GET_PRODUCT_ATAXES to public;
```

1.4.7.3 Text of procedure STP_GET_PRODUCT_ATAXES

```
set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer,
  P_FLAG_ES      smallint,
  P_FLAG_CNS     smallint,
  P_FK_PAIS_ORGM smallint,
  P_FK_PAIS_DSTN smallint,
  P_FK_ESTD_ORGM varchar(2),
  P_FK_ESTD_DSTN varchar(2),
  P_FK_MUNICIPIOS integer,
  P_TYPE_PRINTER integer
)
returns (
  R_RED_BASC      numeric(5, 2),
  R_ALQ_ICMS      numeric(5, 2),
  R_ALQ_ICMSS     numeric(5, 2),
  R_ALQ_IPI       numeric(5, 2),
  R_ALQ_OTR       numeric(5, 2),
  R_COD_ICMS_ECF  smallint,
  R_COD_ICMSS_ECF smallint,
  R_COD_IPI_ECF   smallint,
  R_COD_ISS_ECF   smallint,
  R_COD_OTR_ECF   smallint,
  R_FLAG_SRV      smallint,
  R_STATUS        smallint
)
```

```

)
as
declare variable FkTipoImpostos smallint;
declare variable FlagImpst      smallint;
declare variable PaisDestino    smallint;
declare variable PaisOrigem     smallint;
declare variable EstDestino     varchar(2);
declare variable EstOrigem      varchar(2);
declare variable AlqtSaiCons    numeric(5, 2);
declare variable AlqtSai        numeric(5, 2);
declare variable ArbtIcmss      numeric(5, 2);
declare variable CodAlqtSaiCons smallint;
declare variable CodAlqtSai     smallint;
begin
R_RED_BASC      = 0.0;
R_ALQ_ICMS      = 0.0;
R_ALQ_ICMSS     = 0.0;
R_ALQ_IPI       = 0.0;
R_ALQ_OTR       = 0.0;
R_COD_ICMS_ECF  = 0;
R_COD_ICMSS_ECF = 0;
R_COD_IPI_ECF   = 0;
R_COD_ISS_ECF   = 0;
R_COD_OTR_ECF   = 0;
R_FLAG_SRV      = 0;
R_STATUS        = 0;
/* select fisco info
0 ==> Base para ICMS
1 ==> Base para ICMS Substituicao
2 ==> Base para ICMS Isento
3 ==> Base para IPI
4 ==> Base para ISS
5 ==> Base para Outros
*/
PaisOrigem = P_FK_PAIS_ORGM;
PaisDestino = P_FK_PAIS_DSTN;
EstOrigem = P_FK_ESTD_ORGM;
EstDestino = P_FK_ESTD_DSTN;
for select Pri.FK_TIPO_IMPOSTOS, Tim.FLAG_IMPST, Tim.RED_BASC
from PRODUTOS_IMPOSTOS Pri, TIPO_IMPOSTOS Tim
where Pri.FK_EMPRESAS = :P_FK_EMPRESAS
and Pri.FK_PRODUTOS = :P_FK_PRODUTOS
and Tim.PK_TIPO_IMPOSTOS = Pri.FK_TIPO_IMPOSTOS
into :FkTipoImpostos, :FlagImpst, :R_RED_BASC do
begin
if (P_FLAG_ES = 0) then
select FK_PAISES, FK_ESTADOS /* select country and state of the company */
from EMPRESAS
where PK_EMPRESAS = :P_FK_EMPRESAS
into :PaisDestino, :EstDestino;
else
select FK_PAISES, FK_ESTADOS /* select country and state of the company */
from EMPRESAS
where PK_EMPRESAS = :P_FK_EMPRESAS
into :PaisOrigem, :EstOrigem;
select ALQT_CNSF, ALQT_IMPST, ALQT_ARBT
from ALIQUOTAS
where FK_PAISES__ORGM = :PaisOrigem

```

```
and FK_ESTADOS__ORGM = :EstOrigem
and FK_PAISES__DST = :PaisDestino
and FK_ESTADOS__DST = :EstDestino
and FK_TIPO_IMPOSTOS = :FkTipoImpostos
into :AlqtSaiCons, :AlqtSai, :ArbtIcmss;
if (((AlqtSai is null) or (AlqtSai <= 0)) and
(FlagImpst not in (1, 2))) then
R_STATUS = 30; /* Tax of product not found */
if (FlagImpst is null) then
R_STATUS = 40; /* whithout fiscal classification */
R_ALQ_ICMS = 0;
R_ALQ_ICMSS = 0;
R_ALQ_IPI = 0;
R_ALQ_OTR = 0;
if (FlagImpst = 0) then
if (P_FLAG_CNS = 1) then
R_ALQ_ICMS = AlqtSaiCons;
else
R_ALQ_ICMS = AlqtSai;
if (FlagImpst = 1) then
if (P_FLAG_CNS = 1) then
R_ALQ_ICMSS = AlqtSaiCons;
else
R_ALQ_ICMSS = AlqtSai;
if (FlagImpst = 3) then
R_ALQ_IPI = AlqtSai;
if (FlagImpst = 4) then
R_FLAG_SRV = 1;
if (FlagImpst = 5) then
R_ALQ_OTR = AlqtSai;
if ((P_FLAG_CNS = 1) and (P_TYPE_PRINTER is not null) and
(P_TYPE_PRINTER > 0)) then
begin
select COD_ALQT_CNSF, COD_ALQT
from ALIQUOTAS_IMP_FISCAL
where FK_PAISES__ORGM = :PaisOrigem
and FK_ESTADOS__ORGM = :EstOrigem
and FK_PAISES__DST = :PaisDestino
and FK_ESTADOS__DST = :PaisDestino
and FK_TIPO_IMPOSTOS = :FkTipoImpostos
and PK_ALIQUOTAS_IMP_FISCAL = :P_TYPE_PRINTER
into :CodAlqtSaiCons, :CodAlqtSai;
R_COD_ICMS_ECF = 0;
R_COD_ICMSS_ECF = 0;
R_COD_IPI_ECF = 0;
R_COD_ISS_ECF = 0;
R_COD_OTR_ECF = 0;
if (FlagImpst = 0) then
if (P_FLAG_CNS = 1) then
R_COD_ICMS_ECF = CodAlqtSaiCons;
else
R_COD_ICMS_ECF = CodAlqtSai;
if (FlagImpst = 1) then
if (P_FLAG_CNS = 1) then
R_COD_ICMSS_ECF = CodAlqtSaiCons;
else
R_COD_ICMSS_ECF = CodAlqtSai;
if (FlagImpst = 3) then
```

```

R_COD_IPI_ECF = CodAlqtSai;
if (FlagImpst = 4) then
begin
select COD_ISS_ECF
from MUNICIPIOS_ALIQUOTAS
where FK_PAISES          = :PaisOrigem
and FK_ESTADOS           = :EstOrigem
and FK_MUNICIPIOS        = :P_FK_MUNICIPIOS
and PK_MUNICIPIOS_ALIQUOTAS = :P_TYPE_PRINTER
into :R_COD_ISS_ECF;
end /* end of if */
if (FlagImpst = 5) then
R_COD_OTR_ECF = CodAlqtSai;
end /* end of if */
end
suspend;
end ^

set term ;^

grant execute on procedure %PROC% to public;

```

1.4.7.4 List of diagrams containing the procedure STP_GET_PRODUCT_ATAXES

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.8 Procedure STP_GET_PRODUCT_BOX

1.4.8.1 Card of procedure STP_GET_PRODUCT_BOX

<i>Name</i>	STP_GET_PRODUCT_BOX
<i>Code</i>	STP_GET_PRODUCT_BOX
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.8.2 Code preview of procedure STP_GET_PRODUCT_BOX

```

set term ^;

create procedure STP_GET_PRODUCT_BOX (
P_FK_EMPRESAS integer,
P_PK_PRODUTOS integer,
P_QTD_PROD numeric(9, 4)
)
returns (
R_FK_ALMOXARIFADOS smallint,
R_FK_LOTACOES smallint,
R_QTD_ESTQ_ALMX numeric(11, 4),
R_QTD_ESTQ_LOT numeric(11, 4),
R_LOT_PROD varchar(20)
)
as
declare variable Rua varchar(6);
declare variable Nivel varchar(6);
declare variable Box varchar(6);

```



```

begin
  if (P_QTD_PROD is null) then
    P_QTD_PROD = 0;
  for select (Pes.QTD_ESTQ - (Pes.QTD_GRNT + Pes.QTD_RSRV +
    Pes.QTD_ESTQ_QR + Pes.QTD_GRNT)) as ESTOQUES,
    Pes.FK_ALMOXARIFADOS, Plo.QTD_LOT, Lot.RUA_LOT,
    Lot.NIVEL_LOT, Lot.BOX_LOT, Lot.PK_LOTACOES
  from PRODUTOS_ESTOQUES Pes
  left outer join PRODUTOS_LOTACOES Plo
    on Plo.FK_EMPRESAS = Pes.FK_EMPRESAS
  and Plo.FK_ALMOXARIFADOS = Pes.FK_ALMOXARIFADOS
  and Plo.FK_PRODUTOS = Pes.FK_PRODUTOS
  left outer join LOTACOES Lot
    on Lot.FK_EMPRESAS = Pes.FK_EMPRESAS
  and Lot.FK_ALMOXARIFADOS = Pes.FK_ALMOXARIFADOS
  and Lot.PK_LOTACOES = Plo.FK_LOTACOES
  where Pes.FK_EMPRESAS = :P_FK_EMPRESAS
  and Pes.FK_PRODUTOS = :P_PK_PRODUTOS
  into :R_FK_ALMOXARIFADOS, :R_QTD_ESTQ_ALMX, :R_QTD_ESTQ_LOT, :Rua,
    :Nivel, :Box, :R_FK_LOTACOES do
begin
  if ((P_QTD_PROD = 0) or (P_QTD_PROD <= R_QTD_ESTQ_ALMX)) then
begin
  R_LOT_PROD = Rua || '-' || Nivel || '-' || Box;
  suspend;
end
end
end ^

set term ^;

grant execute on procedure STP_GET_PRODUCT_BOX to public;

```

1.4.8.3 Text of procedure STP_GET_PRODUCT_BOX

```

set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_PK_PRODUTOS integer,
  P_QTD_PROD numeric(9, 4)
)
returns (
  R_FK_ALMOXARIFADOS smallint,
  R_FK_LOTACOES smallint,
  R_QTD_ESTQ_ALMX numeric(11, 4),
  R_QTD_ESTQ_LOT numeric(11, 4),
  R_LOT_PROD varchar(20)
)
as
  declare variable Rua varchar(6);
  declare variable Nivel varchar(6);
  declare variable Box varchar(6);
begin
  if (P_QTD_PROD is null) then
    P_QTD_PROD = 0;
  for select (Pes.QTD_ESTQ - (Pes.QTD_GRNT + Pes.QTD_RSRV +
    Pes.QTD_ESTQ_QR + Pes.QTD_GRNT)) as ESTOQUES,

```

```

    Pes.FK_ALMOXARIFADOS, Plo.QTD_LOT, Lot.RUA_LOT,
    Lot.NIVEL_LOT, Lot.BOX_LOT, Lot.PK_LOTACOES
from PRODUTOS_ESTOQUES Pes
left outer join PRODUTOS_LOTACOES Plo
  on Plo.FK_EMPRESAS = Pes.FK_EMPRESAS
and Plo.FK_ALMOXARIFADOS = Pes.FK_ALMOXARIFADOS
and Plo.FK_PRODUTOS = Pes.FK_PRODUTOS
left outer join LOTACOES Lot
  on Lot.FK_EMPRESAS = Pes.FK_EMPRESAS
and Lot.FK_ALMOXARIFADOS = Pes.FK_ALMOXARIFADOS
and Lot.PK_LOTACOES = Plo.FK_LOTACOES
where Pes.FK_EMPRESAS = :P_FK_EMPRESAS
and Pes.FK_PRODUTOS = :P_PK_PRODUTOS
into :R_FK_ALMOXARIFADOS, :R_QTD_ESTQ_ALMX, :R_QTD_ESTQ_LOT, :Rua,
:Nivel, :Box, :R_FK_LOTACOES do
begin
  if ((P_QTD_PROD = 0) or (P_QTD_PROD <= R_QTD_ESTQ_ALMX)) then
  begin
    R_LOT_PROD = Rua || '-' || Nivel || '-' || Box;
    suspend;
  end
end
end ^

set term ^;

grant execute on procedure %PROC% to public;

```

1.4.8.4 List of diagrams containing the procedure STP_GET_PRODUCT_BOX

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.9 Procedure STP_GET_PRODUCT_ITEM

1.4.9.1 Card of procedure STP_GET_PRODUCT_ITEM

<i>Name</i>	STP_GET_PRODUCT_ITEM
<i>Code</i>	STP_GET_PRODUCT_ITEM
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.9.2 Code preview of procedure STP_GET_PRODUCT_ITEM

```

set term ^;

create procedure STP_GET_PRODUCT_ITEM (
  P_FK_EMPRESA      integer,
  P_FK_CADASTRO     integer,
  P_FK_PRODUTO      varchar(30),
  P_CODE_TYPE       smallint,
  P_FK_TABELA_PRECOS smallint,
  P_FK_GRUPOS_MOVIMENTOS smallint,
  P_FK_TIPO_MOVIMENTOS smallint,
  P_TYPE_VOLUME     smallint,
  P_QTD_PROD        numeric(18, 4)
)

```

```

returns (
  R_PK_PRODUTOS      integer,
  R_FK_TABELA_PRECOS smallint,
  R_DSC_TAB          varchar(30),
  R_COD_REF          varchar(20),
  R_COD_BARRA        varchar(30),
  R_DSC_PROD         varchar(50),
  R_DSC_RED          varchar(30),
  R_LOT_PROD         varchar(20),
  R_QTD_ESTQ         numeric(9, 4),
  R_QTD_ESTQ_ALMX    numeric(9, 4),
  R_QTD_ESTQ_LOT     numeric(9, 4),
  R_QTD_ITEM         numeric(9, 4),
  R_MNM_UNI          varchar(2),
  R_VLR_TAB          numeric(9, 4),
  R_VLR_UNIT         numeric(9, 4),
  R_VLR_DSCT         numeric(9, 4),
  R_DSCT_ITEM        numeric(9, 4),
  R_PES_LIQ          numeric(9, 4),
  R_PES_BRU          numeric(9, 4),
  R_VOL_PROD         double precision,
  R_SIT_TRIB         varchar(3),
  R_CLASS_FISCAL     varchar(20),
  R_ALQ_OTR          numeric(5, 2),
  R_ALQ_ISS          numeric(5, 2),
  R_ALQ_ICMS         numeric(5, 2),
  R_ALQ_ICMSS        numeric(5, 2),
  R_COD_ICMS_ECF     varchar(3),
  R_COD_ICMSS_ECF    varchar(3),
  R_COD_IPI_ECF      varchar(3),
  R_COD_ISS_ECF      varchar(3),
  R_COD_OTR_ECF      varchar(3),
  R_ALQ_IPI          numeric(5, 2),
  R_FLAG_FRAC        smallint,
  R_FLAG_SRV         smallint,
  R_FLAG_DEFTAB      smallint,
  R_FLAG_TAB         smallint,
  R_COM_PROD         numeric(5, 2),
  R_RED_BASC         numeric(5, 2),
  R_FLAG_CONF        smallint,
  R_STATUS           smallint,
  R_FK_UNIDADES      smallint,
  R_FK_ALMOXARIFADOS smallint,
  R_FK_LOTACOES      smallint,
  R_PERC_DSCT        numeric(5, 2),
  R_FK_EMPRESAS      integer,
  R_NOM_EMP          varchar(50)
)

```

```

)
as

```

```

declare variable FkSecoes      integer;
declare variable FkGrupos      integer;
declare variable FkSubGrupos   integer;
declare variable FlagES        smallint;
declare variable FlagCns       smallint;
declare variable FlagExp       smallint;
declare variable FlagCusto     smallint;
declare variable PaisOrigem    smallint;
declare variable EstOrigem     char(2);

```

```
declare variable PaisDestino    smallint;
declare variable EstDestino     char(2);
declare variable ComSGru        numeric(5, 2);
declare variable PkTabelaPrecos smallint;
declare variable VlrTab         numeric(11, 4);
declare variable CustFinal      numeric(09, 4);
declare variable CustForn       numeric(09, 4);
declare variable CustReal       numeric(09, 4);
declare variable CustMed        numeric(09, 4);
declare variable FlagPrm        smallint;
declare variable DscTab         varchar(30);
declare variable DscAux         varchar(50);
declare variable FlagFix        smallint;
declare variable Height         numeric(09, 4);
declare variable Width          numeric(09, 4);
declare variable Depth          numeric(09, 4);
declare variable FkLanguage     varchar(05);
declare variable FkMunicipios   integer;
declare variable FlagBoxSus     smallint;

begin
  R_STATUS = 0; /* All conditions was attended */
  R_FLAG_SRV = 0;
  if ((P_QTD_PROD is null) or (P_QTD_PROD = 0)) then
    R_QTD_ITEM = 1;
  else
    R_QTD_ITEM = :P_QTD_PROD;
  if ((P_FK_TABELA_PRECOS = 0) or (P_FK_TABELA_PRECOS is null)) then
  begin
    R_STATUS = -5; /* I can't to calc the price */
    suspend;
    exit;
  end /* end of if */
  /* select product info */
  select Prc.FK_PRODUTOS, Prc.COD_REF, Prd.FK_UNIDADES, Prd.DSC_PROD,
    Prd.FK_SECOES, Prd.FK_GRUPOS, Prd.FK_SUBGRUPOS, Uni.MNMO_UNI,
    Uni.FLAG_FRAC
  from PRODUTOS_CODIGOS Prc, PRODUTOS Prd, UNIDADES Uni
  where Prc.COD_REF = :P_FK_PRODUTO
  and Prc.FLAG_TCODE = :P_CODE_TYPE
  and Prd.PK_PRODUTOS = Prc.FK_PRODUTOS
  and Uni.PK_UNIDADES = Prd.FK_UNIDADES
  into :R_PK_PRODUTOS, :R_COD_REF, :R_FK_UNIDADES, :R_DSC_PROD,
    :FkSecoes, :FkGrupos, :FkSubGrupos, :R_MNM_UNI,
    :R_FLAG_FRAC;
  if ((:R_COD_REF is null) or (R_COD_REF = '')) then
  begin
    R_STATUS = -10; /* Product not Found */
    suspend;
    exit;
  end /* end of if */
  if (P_CODE_TYPE = 2) then
  begin
    R_COD_BARRA = R_COD_REF;
    select COD_REF
    from PRODUTOS_CODIGOS Prc, PRODUTOS Prd
    where Prc.FK_PRODUTOS = :R_PK_PRODUTOS
    and Prc.FLAG_TCODE = 2
    into :R_COD_REF;
```

```

end /* end of if */
else
  R_COD_BARRA = "";
select Gmv.FLAG_ES, Tmv.FLAG_CNS, Tmv.FLAG_EXP /* select type of customer and Type of movemementation*/
from GRUPOS_MOVIMENTOS Gmv, TIPO_MOVIMENTOS Tmv
where Gmv.PK_GRUPOS_MOVIMENTOS = :P_FK_GRUPOS_MOVIMENTOS
  and Tmv.FK_GRUPOS_MOVIMENTOS = Gmv.PK_GRUPOS_MOVIMENTOS
  and Tmv.PK_TIPO_MOVIMENTOS = :P_FK_TIPO_MOVIMENTOS
into :FlagES, :FlagCns, :FlagExp;
if (:FlagES is null) then
begin
  R_STATUS = -20; /* Movemementation not found */
  suspend;
  exit;
end /* end of if */
/* select language, country and state from customer */
if (FlagES = 0) then
begin
  select Cad.FK_PAISES, Cad.FK_ESTADOS, Pais.FK_LINGUAGENS,
    Mun.ALQ_ISS, Mun.PK_MUNICIPIOS
  from CADASTROS Cad, PAISES Pais, MUNICIPIOS Mun
  where Cad.PK_CADASTROS = :P_FK_CADASTRO
    and Pais.PK_PAISES = Cad.FK_PAISES
    and Mun.FK_PAISES = Cad.FK_PAISES
    and Mun.FK_ESTADOS = Cad.FK_ESTADOS
    and Mun.PK_MUNICIPIOS = Cad.FK_MUNICIPIOS
  into :PaisOrigem, :EstOrigem, :FkLanguage,
    :R_ALQ_ISS, :FkMunicipios;
end /* end of if */
else
begin
  select Cad.FK_PAISES, Cad.FK_ESTADOS, Pais.FK_LINGUAGENS,
    Mun.ALQ_ISS, Mun.PK_MUNICIPIOS
  from CADASTROS Cad, PAISES Pais, MUNICIPIOS Mun
  where Cad.PK_CADASTROS = :P_FK_CADASTRO
    and Pais.PK_PAISES = Cad.FK_PAISES
    and Mun.FK_PAISES = Cad.FK_PAISES
    and Mun.FK_ESTADOS = Cad.FK_ESTADOS
    and Mun.PK_MUNICIPIOS = Cad.FK_MUNICIPIOS
  into :PaisDestino, :EstDestino, :FkLanguage,
    :R_ALQ_ISS, :FkMunicipios;
end /* end of else */
if (R_ALQ_ISS is null) then R_ALQ_ISS = 0.0;
/* select description from customer language */
select DSC_PROD, DSC_PROD_RED
from DESCRICOES_PRD_LANG
where FK_LINGUAGENS = :FkLanguage
  and FK_PRODUTOS = :R_PK_PRODUTOS
into :DscAux, :R_DSC_RED;
if ((DscAux is not null) and (DscAux <> "")) then
  R_DSC_PROD = DscAux;
else
  R_DSC_RED = "";
if (FlagES = 1) then
begin
  select Prv.PES_LIQ, Prv.PES_BRU, Lin.COM_LIN /* Get Comission and Peso of Product */
  from PRODUTOS_VENDAS Prv, LINHAS Lin
  where Prv.FK_PRODUTOS = :R_PK_PRODUTOS

```

```

and PK_LINHAS = Prv.FK_LINHAS
into :R_PES_LIQ, :R_PES_BRU, :R_COM_PROD;
/* Calc volume of the product */
select ALT_EPROD, PROF_EPROD, LARG_EPROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
if (P_TYPE_VOLUME = 1) then
select ALT_IPROD, PROF_IPROD, LARG_IPROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
if (P_TYPE_VOLUME = 2) then
select ALT_PROD, PROF_PROD, LARG_PROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
R_VOL_PROD = Height * Width * Depth;
end
/* select supply for the product */
for select (Pct.QTD_ESTQ - (Pct.QTD_GRNT + Pct.QTD_RSRV +
Pct.QTD_ESTQ_QR + Pct.QTD_GRNT)) as ESTOQUES,
Pct.CUST_FINAL, Pct.CUST_FORN, Pct.CUST_MED,
Pct.CUST_REAL, FK_EMPRESAS
from PRODUTOS_CUSTOS Pct
where ((Pct.FK_EMPRESAS = :P_FK_EMPRESA)
or (-1 = :P_FK_EMPRESA))
and Pct.FK_PRODUTOS = :R_PK_PRODUTOS
into :R_QTD_ESTQ, :CustFinal, CustForn, CustMed,
:CustReal, :R_FK_EMPRESAS do
begin
/* Select name of active company */
select RAZ_SOC from EMPRESAS
where PK_EMPRESAS = :R_FK_EMPRESAS
into :R_NOM_EMP;
/* Verify if the product has a configuration */
select count(*) from ACABAMENTO_PRECOS
where FK_EMPRESAS = :R_FK_EMPRESAS
and FK_PRODUTOS = :R_PK_PRODUTOS
into :R_FLAG_CONF;
if (R_FLAG_CONF is null) then
R_FLAG_CONF = 0;
if (R_FLAG_CONF > 1) then
R_FLAG_CONF = 1;
select R_RED_BASC, R_ALQ_ICMS, R_ALQ_ICMSS, R_ALQ_IPI, R_ALQ_OTR,
R_COD_ICMS_ECF, R_COD_ICMSS_ECF, R_COD_IPI_ECF, R_COD_ISS_ECF,
R_COD_OTR_ECF, R_FLAG_SRV, R_STATUS
from STP_GET_PRODUCT_ATADES (:R_FK_EMPRESAS, :R_PK_PRODUTOS, :FlagES,
:FlagCns, :PaisOrigem, :PaisDestino, :EstOrigem, :EstDestino,
:FkMunicipios, 0 /* Set here the flag that indicate the fiscal printer */)
into :R_RED_BASC, :R_ALQ_ICMS, :R_ALQ_ICMSS, :R_ALQ_IPI, :R_ALQ_OTR,
:R_COD_ICMS_ECF, :R_COD_ICMSS_ECF, :R_COD_IPI_ECF, :R_COD_ISS_ECF,
:R_COD_OTR_ECF, :R_FLAG_SRV, :R_STATUS;
if (FlagES = 1) then
begin
/* select price info */
R_FK_TABELA_PRECOS = 0;
R_VLR_TAB = 0;

```

```
select Pmr.SIT_TRIB, Pim.FK_CLASSIFICACAO_FISCAL
  from PRODUTOS_MARGEM Pmr, PRODUTOS_IMPOSTOS Pim
 where Pmr.FK_EMPRESAS = :R_FK_EMPRESAS
    and Pmr.FK_PRODUTOS = :R_PK_PRODUTOS
    and Pim.FK_EMPRESAS = Pmr.FK_EMPRESAS
    and Pim.FK_PRODUTOS = Pmr.FK_PRODUTOS
 into :R_SIT_TRIB, :R_CLASS_FISCAL;
select Tpr.PK_TABELA_PRECOS, Prp.PRE_VDA, Tpr.DSC_TAB,
      Prp.FLAG_FIX, Tpr.PERC_DSCT, Tpr.FLAG_DEFTAB
  from PRODUTOS_PRECOS Prp, TABELA_PRECOS Tpr
 where Tpr.DTA_INI      <= current_date
    and ((Tpr.DTA_FIN    >= current_date)
    or ( Tpr.DTA_FIN     is null))
    and Prp.FK_EMPRESAS  = :R_FK_EMPRESAS
    and Prp.FK_PRODUTOS  = :R_PK_PRODUTOS
    and Prp.FK_TABELA_PRECOS = Tpr.PK_TABELA_PRECOS
 into :PkTabelaPrecos, :VlrTab, :DscTab, :FlagFix,
      :R_PERC_DSCT, :R_FLAG_DEFTAB;
if (R_PERC_DSCT is null) then
  R_PERC_DSCT = 0;
R_FK_TABELA_PRECOS = PkTabelaPrecos;
R_VLR_TAB          = VlrTab;
R_DSC_TAB          = DscTab;
if (P_FK_TABELA_PRECOS = PkTabelaPrecos) then
  R_FLAG_TAB = 1;
else
  R_FLAG_TAB = 0;
/* calc Discounts if FlagFix = 0 */
if (FlagFix = 0) then
begin
  /* select discount for product group */
  select DSCT_PROD, COM_SGRU
    from SUBGRUPOS
   where FK_SECOES   = :FkSecoes
     and FK_GRUPOS   = :FkGrupos
     and PK_SUBGRUPOS = :FkSubGrupos
   into :R_DSCT_ITEM, :ComSgru;
  if (R_DSCT_ITEM is null) then
    R_DSCT_ITEM = 0;
  /* Calc of the unit value with discount */
  VlrTab = R_VLR_TAB;
  if (R_DSCT_ITEM > 0) then
    VlrTab = VlrTab - ((VlrTab * R_DSCT_ITEM) / 100);
  if ((R_PERC_DSCT > 0) and (FlagPrm > 0)) then
    VlrTab = VlrTab - ((VlrTab * R_PERC_DSCT) / 100);
  R_VLR_UNIT   = VlrTab;
  R_VLR_DSCT   = R_VLR_TAB - R_VLR_UNIT;
end
end
if ((R_SIT_TRIB is null) or (R_SIT_TRIB = '')) then
  R_STATUS = 30; /* Taxes not found */
if (:FlagES = 0) then
begin
  R_VLR_TAB = 0;
  R_VLR_UNIT = CustReal;
  select FLAG_TCUSTO from PARAMETROS_ESTQ
   where FK_EMPRESAS = :R_FK_EMPRESAS
  into :FlagCusto;
```

```

/*
0 ==> Custo Medio
1 ==> Custo Real (PMZ)
2 ==> Custo Fornecedor Final
3 ==> Custo Fornecedor Tabela
4 ==> Custo Referencia
*/
if (FlagCusto = 0) then
    R_VLR_TAB = CustMed;
if (FlagCusto = 1) then
    R_VLR_TAB = CustReal;
if (FlagCusto = 2) then
    R_VLR_TAB = CustForn; /* verificar o que incide sobre este custo */
if (FlagCusto = 3) then
    R_VLR_TAB = CustForn;
if (FlagCusto = 4) then
    R_VLR_TAB = CustFinal;
end
FlagBoxSus = 0;
R_QTD_ESTQ_ALMX = 0.0;
R_QTD_ESTQ_LOT = 0.0;
R_LOT_PROD = "";
for select R_QTD_ESTQ_ALMX, R_QTD_ESTQ_LOT, R_LOT_PROD,
    R_FK_LOTACOES, R_FK_ALMOXARIFADOS
    from STP_GET_PRODUCT_BOX (:R_FK_EMPRESAS, :R_PK_PRODUTOS, 0)
    into :R_QTD_ESTQ_ALMX, :R_QTD_ESTQ_LOT, :R_LOT_PROD,
        :R_FK_LOTACOES, :R_FK_ALMOXARIFADOS do
begin
    suspend;
    FlagBoxSus = 1;
end
if (FlagBoxSus = 0) then
    suspend;
end
if (R_QTD_ESTQ is null) then
begin
    R_STATUS = -50; /* whithout Stoks and Cost */
    suspend;
end
end ^

set term ^;

```

grant EXECUTE on procedure STP_GET_PRODUCT_ITEM to PUBLIC;

1.4.9.3 Text of procedure STP_GET_PRODUCT_ITEM

set term ^;

```

create procedure %PROC% (
    P_FK_EMPRESA      integer,
    P_FK_CADASTRO     integer,
    P_FK_PRODUTO      varchar(30),
    P_CODE_TYPE       smallint,
    P_FK_TABELA_PRECOS smallint,
    P_FK_GRUPOS_MOVIMENTOS smallint,
    P_FK_TIPO_MOVIMENTOS smallint,
    P_TYPE_VOLUME     smallint,

```



```

P_QTD_PROD      numeric(18, 4)
)
returns (
R_PK_PRODUTOS   integer,
R_FK_TABELA_PRECOS smallint,
R_DSC_TAB       varchar(30),
R_COD_REF       varchar(20),
R_COD_BARRA     varchar(30),
R_DSC_PROD      varchar(50),
R_DSC_RED       varchar(30),
R_LOT_PROD      varchar(20),
R_QTD_ESTQ      numeric(9, 4),
R_QTD_ESTQ_ALMX numeric(9, 4),
R_QTD_ESTQ_LOT  numeric(9, 4),
R_QTD_ITEM      numeric(9, 4),
R_MNM_UNI       varchar(2),
R_VLR_TAB       numeric(9, 4),
R_VLR_UNIT      numeric(9, 4),
R_VLR_DSCT      numeric(9, 4),
R_DSCT_ITEM     numeric(9, 4),
R_PES_LIQ       numeric(9, 4),
R_PES_BRU       numeric(9, 4),
R_VOL_PROD      double precision,
R_SIT_TRIB      varchar(3),
R_CLASS_FISCAL  varchar(20),
R_ALQ_OTR       numeric(5, 2),
R_ALQ_ISS       numeric(5, 2),
R_ALQ_ICMS      numeric(5, 2),
R_ALQ_ICMSS     numeric(5, 2),
R_COD_ICMS_ECF  varchar(3),
R_COD_ICMSS_ECF varchar(3),
R_COD_IPI_ECF   varchar(3),
R_COD_ISS_ECF   varchar(3),
R_COD_OTR_ECF   varchar(3),
R_ALQ_IPI       numeric(5, 2),
R_FLAG_FRAC     smallint,
R_FLAG_SRV      smallint,
R_FLAG_DEFTAB   smallint,
R_FLAG_TAB      smallint,
R_COM_PROD      numeric(5, 2),
R_RED_BASC      numeric(5, 2),
R_FLAG_CONF     smallint,
R_STATUS        smallint,
R_FK_UNIDADES   smallint,
R_FK_ALMOXARIFADOS smallint,
R_FK_LOTACOES   smallint,
R_PERC_DSCT     numeric(5, 2),
R_FK_EMPRESAS   integer,
R_NOM_EMP       varchar(50)
)
as
declare variable FkSecoes   integer;
declare variable FkGrupos   integer;
declare variable FkSubGrupos integer;
declare variable FlagES     smallint;
declare variable FlagCns    smallint;
declare variable FlagExp    smallint;
declare variable FlagCusto  smallint;

```

```
declare variable PaisOrigem    smallint;
declare variable EstOrigem     char(2);
declare variable PaisDestino   smallint;
declare variable EstDestino    char(2);
declare variable ComSGru       numeric(5, 2);
declare variable PkTabelaPrecos smallint;
declare variable VlrTab        numeric(11, 4);
declare variable CustFinal     numeric(09, 4);
declare variable CustForn      numeric(09, 4);
declare variable CustReal      numeric(09, 4);
declare variable CustMed       numeric(09, 4);
declare variable FlagPrm       smallint;
declare variable DscTab        varchar(30);
declare variable DscAux        varchar(50);
declare variable FlagFix       smallint;
declare variable Height        numeric(09, 4);
declare variable Width         numeric(09, 4);
declare variable Depth         numeric(09, 4);
declare variable FkLanguage    varchar(05);
declare variable FkMunicipios  integer;
declare variable FlagBoxSus    smallint;

begin
  R_STATUS = 0; /* All conditions was attended */
  R_FLAG_SRV = 0;
  if ((P_QTD_PROD is null) or (P_QTD_PROD = 0)) then
    R_QTD_ITEM = 1;
  else
    R_QTD_ITEM = :P_QTD_PROD;
  if ((P_FK_TABELA_PRECOS = 0) or (P_FK_TABELA_PRECOS is null)) then
    begin
      R_STATUS = -5; /* I can't to calc the price */
      suspend;
      exit;
    end /* end of if */
  /* select product info */
  select Prc.FK_PRODUTOS, Prc.COD_REF, Prd.FK_UNIDADES, Prd.DSC_PROD,
    Prd.FK_SECOES, Prd.FK_GRUPOS, Prd.FK_SUBGRUPOS, Uni.MNMO_UNI,
    Uni.FLAG_FRAC
    from PRODUTOS_CODIGOS Prc, PRODUTOS Prd, UNIDADES Uni
  where Prc.COD_REF = :P_FK_PRODUTO
    and Prc.FLAG_TCODE = :P_CODE_TYPE
    and Prd.PK_PRODUTOS = Prc.FK_PRODUTOS
    and Uni.PK_UNIDADES = Prd.FK_UNIDADES
  into :R_PK_PRODUTOS, :R_COD_REF, :R_FK_UNIDADES, :R_DSC_PROD,
    :FkSecoes, :FkGrupos, :FkSubGrupos, :R_MNM_UNI,
    :R_FLAG_FRAC;
  if ((:R_COD_REF is null) or (R_COD_REF = '')) then
    begin
      R_STATUS = -10; /* Product not Found */
      suspend;
      exit;
    end /* end of if */
  if (P_CODE_TYPE = 2) then
    begin
      R_COD_BARRA = R_COD_REF;
      select COD_REF
        from PRODUTOS_CODIGOS Prc, PRODUTOS Prd
       where Prc.FK_PRODUTOS = :R_PK_PRODUTOS
```

```
and Prc.FLAG_TCODE = 2
into :R_COD_REF;
end /* end of if */
else
  R_COD_BARRA = "";
select Gmv.FLAG_ES, Tmv.FLAG_CNS, Tmv.FLAG_EXP /* select type of customer and Type of movementation*/
from GRUPOS_MOVIMENTOS Gmv, TIPO_MOVIMENTOS Tmv
where Gmv.PK_GRUPOS_MOVIMENTOS = :P_FK_GRUPOS_MOVIMENTOS
and Tmv.FK_GRUPOS_MOVIMENTOS = Gmv.PK_GRUPOS_MOVIMENTOS
and Tmv.PK_TIPO_MOVIMENTOS = :P_FK_TIPO_MOVIMENTOS
into :FlagES, :FlagCns, :FlagExp;
if (:FlagES is null) then
begin
  R_STATUS = -20; /* Movementation not found */
  suspend;
  exit;
end /* end of if */
/* select language, country and state from customer */
if (FlagES = 0) then
begin
  select Cad.FK_PAISES, Cad.FK_ESTADOS, Pais.FK_LINGUAGENS,
    Mun.ALQ_ISS, Mun.PK_MUNICIPIOS
  from CADASTROS Cad, PAISES Pais, MUNICIPIOS Mun
  where Cad.PK_CADASTROS = :P_FK_CADASTRO
  and Pais.PK_PAISES = Cad.FK_PAISES
  and Mun.FK_PAISES = Cad.FK_PAISES
  and Mun.FK_ESTADOS = Cad.FK_ESTADOS
  and Mun.PK_MUNICIPIOS = Cad.FK_MUNICIPIOS
  into :PaisOrigem, :EstOrigem, :FkLanguage,
    :R_ALQ_ISS, :FkMunicipios;
end /* end of if */
else
begin
  select Cad.FK_PAISES, Cad.FK_ESTADOS, Pais.FK_LINGUAGENS,
    Mun.ALQ_ISS, Mun.PK_MUNICIPIOS
  from CADASTROS Cad, PAISES Pais, MUNICIPIOS Mun
  where Cad.PK_CADASTROS = :P_FK_CADASTRO
  and Pais.PK_PAISES = Cad.FK_PAISES
  and Mun.FK_PAISES = Cad.FK_PAISES
  and Mun.FK_ESTADOS = Cad.FK_ESTADOS
  and Mun.PK_MUNICIPIOS = Cad.FK_MUNICIPIOS
  into :PaisDestino, :EstDestino, :FkLanguage,
    :R_ALQ_ISS, :FkMunicipios;
end /* end of else */
if (R_ALQ_ISS is null) then R_ALQ_ISS = 0.0;
/* select description from customer language */
select DSC_PROD, DSC_PROD_RED
from DESCRICOES_PRD_LANG
where FK_LINGUAGENS = :FkLanguage
and FK_PRODUTOS = :R_PK_PRODUTOS
into :DscAux, :R_DSC_RED;
if ((DscAux is not null) and (DscAux <> "")) then
  R_DSC_PROD = DscAux;
else
  R_DSC_RED = "";
if (FlagES = 1) then
begin
  select Prv.PES_LIQ, Prv.PES_BRU, Lin.COM_LIN /* Get Comission and Peso of Product */
```

```
from PRODUTOS_VENDAS Prv, LINHAS Lin
where Prv.FK_PRODUTOS = :R_PK_PRODUTOS
and PK_LINHAS = Prv.FK_LINHAS
into :R_PES_LIQ, :R_PES_BRU, :R_COM_PROD;
/* Calc volume of the product */
select ALT_EPROD, PROF_EPROD, LARG_EPROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
if (P_TYPE_VOLUME = 1) then
select ALT_IPROD, PROF_IPROD, LARG_IPROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
if (P_TYPE_VOLUME = 2) then
select ALT_PROD, PROF_PROD, LARG_PROD
from PRODUTOS_VENDAS
where FK_PRODUTOS = :R_PK_PRODUTOS
into :Height, :Width, :Depth;
R_VOL_PROD = Height * Width * Depth;
end
/* select supply for the product */
for select (Pct.QTD_ESTQ - (Pct.QTD_GRNT + Pct.QTD_RSRV +
Pct.QTD_ESTQ_QR + Pct.QTD_GRNT)) as ESTOQUES,
Pct.CUST_FINAL, Pct.CUST_FORN, Pct.CUST_MED,
Pct.CUST_REAL, FK_EMPRESAS
from PRODUTOS_CUSTOS Pct
where ((Pct.FK_EMPRESAS = :P_FK_EMPRESA)
or (-1 = :P_FK_EMPRESA))
and Pct.FK_PRODUTOS = :R_PK_PRODUTOS
into :R_QTD_ESTQ, :CustFinal, CustForn, CustMed,
:CustReal, :R_FK_EMPRESAS do
begin
/* Select name of active company */
select RAZ_SOC from EMPRESAS
where PK_EMPRESAS = :R_FK_EMPRESAS
into :R_NOM_EMP;
/* Verify if the product has a configuration */
select count(*) from ACABAMENTO_PRECOS
where FK_EMPRESAS = :R_FK_EMPRESAS
and FK_PRODUTOS = :R_PK_PRODUTOS
into :R_FLAG_CONF;
if (R_FLAG_CONF is null) then
R_FLAG_CONF = 0;
if (R_FLAG_CONF > 1) then
R_FLAG_CONF = 1;
select R_RED_BASC, R_ALQ_ICMS, R_ALQ_ICMSS, R_ALQ_IPI, R_ALQ_OTR,
R_COD_ICMS_ECF, R_COD_ICMSS_ECF, R_COD_IPI_ECF, R_COD_ISS_ECF,
R_COD_OTR_ECF, R_FLAG_SRV, R_STATUS
from STP_GET_PRODUCT_ATAxes (:R_FK_EMPRESAS, :R_PK_PRODUTOS, :FlagES,
:FlagCns, :PaisOrigem, :PaisDestino, :EstOrigem, :EstDestino,
:FkMunicipios, 0 /* Set here the flag that indicate the fiscal printer */)
into :R_RED_BASC, :R_ALQ_ICMS, :R_ALQ_ICMSS, :R_ALQ_IPI, :R_ALQ_OTR,
:R_COD_ICMS_ECF, :R_COD_ICMSS_ECF, :R_COD_IPI_ECF, :R_COD_ISS_ECF,
:R_COD_OTR_ECF, :R_FLAG_SRV, :R_STATUS;
if (FlagES = 1) then
begin
/* select price info */
```

```
R_FK_TABELA_PRECOS = 0;
R_VLR_TAB = 0;
select Pmr.SIT_TRIB, Pim.FK_CLASSIFICACAO_FISCAL
  from PRODUTOS_MARGEM Pmr, PRODUTOS_IMPOSTOS Pim
 where Pmr.FK_EMPRESAS = :R_FK_EMPRESAS
    and Pmr.FK_PRODUTOS = :R_PK_PRODUTOS
    and Pim.FK_EMPRESAS = Pmr.FK_EMPRESAS
    and Pim.FK_PRODUTOS = Pmr.FK_PRODUTOS
 into :R_SIT_TRIB, :R_CLASS_FISCAL;
select Tpr.PK_TABELA_PRECOS, Prp.PRE_VDA, Tpr.DSC_TAB,
  Prp.FLAG_FIX, Tpr.PERC_DSCT, Tpr.FLAG_DEFTAB
  from PRODUTOS_PRECOS Prp, TABELA_PRECOS Tpr
 where Tpr.DTA_INI <= current_date
    and ((Tpr.DTA_FIN >= current_date)
    or ( Tpr.DTA_FIN is null))
    and Prp.FK_EMPRESAS = :R_FK_EMPRESAS
    and Prp.FK_PRODUTOS = :R_PK_PRODUTOS
    and Prp.FK_TABELA_PRECOS = Tpr.PK_TABELA_PRECOS
 into :PkTabelaPrecos, :VlrTab, :DscTab, :FlagFix,
    :R_PERC_DSCT, :R_FLAG_DEFTAB;
if (R_PERC_DSCT is null) then
  R_PERC_DSCT = 0;
R_FK_TABELA_PRECOS = PkTabelaPrecos;
R_VLR_TAB = VlrTab;
R_DSC_TAB = DscTab;
if (P_FK_TABELA_PRECOS = PkTabelaPrecos) then
  R_FLAG_TAB = 1;
else
  R_FLAG_TAB = 0;
/* calc Discounts if FlagFix = 0 */
if (FlagFix = 0) then
begin
  /* select discount for product group */
  select DSCT_PROD, COM_SGRU
    from SUBGRUPOS
   where FK_SECOES = :FkSecoes
     and FK_GRUPOS = :FkGrupos
     and PK_SUBGRUPOS = :FkSubGrupos
   into :R_DSCT_ITEM, :ComSgru;
  if (R_DSCT_ITEM is null) then
    R_DSCT_ITEM = 0;
  /* Calc of the unit value with discount */
  VlrTab = R_VLR_TAB;
  if (R_DSCT_ITEM > 0) then
    VlrTab = VlrTab - ((VlrTab * R_DSCT_ITEM) / 100);
  if ((R_PERC_DSCT > 0) and (FlagPrm > 0)) then
    VlrTab = VlrTab - ((VlrTab * R_PERC_DSCT) / 100);
  R_VLR_UNIT = VlrTab;
  R_VLR_DSCT = R_VLR_TAB - R_VLR_UNIT;
end
end
if ((R_SIT_TRIB is null) or (R_SIT_TRIB = '')) then
  R_STATUS = 30; /* Taxes not found */
if (:FlagES = 0) then
begin
  R_VLR_TAB = 0;
  R_VLR_UNIT = CustReal;
  select FLAG_TCUSTO from PARAMETROS_ESTQ
```

```

where FK_EMPRESAS = :R_FK_EMPRESAS
into :FlagCusto;
/*
0 ==> Custo Medio
1 ==> Custo Real (PMZ)
2 ==> Custo Fornecedor Final
3 ==> Custo Fornecedor Tabela
4 ==> Custo Referencia
*/
if (FlagCusto = 0) then
  R_VLR_TAB = CustMed;
if (FlagCusto = 1) then
  R_VLR_TAB = CustReal;
if (FlagCusto = 2) then
  R_VLR_TAB = CustForn; /* verificar o que incide sobre este custo */
if (FlagCusto = 3) then
  R_VLR_TAB = CustForn;
if (FlagCusto = 4) then
  R_VLR_TAB = CustFinal;
end
FlagBoxSus = 0;
R_QTD_ESTQ_ALMX = 0.0;
R_QTD_ESTQ_LOT = 0.0;
R_LOT_PROD = "";
for select R_QTD_ESTQ_ALMX, R_QTD_ESTQ_LOT, R_LOT_PROD,
  R_FK_LOTACOES, R_FK_ALMOXARIFADOS
from STP_GET_PRODUCT_BOX (:R_FK_EMPRESAS, :R_PK_PRODUTOS, 0)
into :R_QTD_ESTQ_ALMX, :R_QTD_ESTQ_LOT, :R_LOT_PROD,
  :R_FK_LOTACOES, :R_FK_ALMOXARIFADOS do
begin
  suspend;
  FlagBoxSus = 1;
end
if (FlagBoxSus = 0) then
  suspend;
end
if (R_QTD_ESTQ is null) then
begin
  R_STATUS = -50; /* whithout Stoks and Cost */
  suspend;
end
end ^

set term ;^

```

1.4.9.4 List of diagrams containing the procedure STP_GET_PRODUCT_ITEM

Name	Code
SysPed	SYSPED

1.4.9.5 List of permissions of the procedure STP_GET_PRODUCT_ITEM

User	Grant	Grant With Option	Revoke	Revoke With Cascade
PUBLIC	EXECUTE			

1.4.10 Procedure STP_PROMOVE_PEDIDO**1.4.10.1 Card of procedure STP_PROMOVE_PEDIDO**

<i>Name</i>	STP_PROMOVE_PEDIDO
<i>Code</i>	STP_PROMOVE_PEDIDO
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.10.2 Code preview of procedure STP_PROMOVE_PEDIDO

set term ^;

```
create procedure STP_PROMOVE_PEDIDO (
  P_FK_EMPRESAS      integer,
  P_FK_TIPO_PEDIDOS  integer,
  P_PK_PEDIDOS        integer,
  P_FK_TIPO_PEDIDOS_NEW integer)
returns (
  R_PK_PEDIDOS integer
)
as
  declare variable FkTipoDocumentos smallint;
begin
  R_PK_PEDIDOS = 0;
  select FK_TIPO_DOCUMENTOS from TIPO_PEDIDOS
  where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS_NEW
  into :FkTipoDocumentos;
  if ((FkTipoDocumentos is not null) and (FkTipoDocumentos > 0)) then
  begin
    select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO
    where FK_EMPRESAS      = :P_FK_EMPRESAS
    and FK_TIPO_DOCUMENTOS = :FkTipoDocumentos
    into :R_PK_PEDIDOS;
    if (R_PK_PEDIDOS is null) then
    begin
      insert into TIPO_DOCUMENTOS_NUMERACAO
      (FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)
      values
      (:P_FK_EMPRESAS, :FkTipoDocumentos, 0);
      R_PK_PEDIDOS = 0;
    end
    R_PK_PEDIDOS = R_PK_PEDIDOS + 1;
    update TIPO_DOCUMENTOS_NUMERACAO set
      NUM_DOC      = :R_PK_PEDIDOS
    where FK_EMPRESAS      = :P_FK_EMPRESAS
    and FK_TIPO_DOCUMENTOS = :FkTipoDocumentos;
  end
  if (R_PK_PEDIDOS > 0) then
  begin
    update PEDIDOS set
      PK_PEDIDOS      = :R_PK_PEDIDOS,
      FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS_NEW
    where FK_EMPRESAS      = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and PK_PEDIDOS      = :P_PK_PEDIDOS;
  end
end
```

```
suspend;  
end ^
```

```
set term ^;
```

```
grant execute on procedure STP_PROMOVE_PEDIDO to public;
```

1.4.10.3 Text of procedure STP_PROMOVE_PEDIDO

```
set term ^;
```

```
create procedure %PROC% (  
  P_FK_EMPRESAS      integer,  
  P_FK_TIPO_PEDIDOS  integer,  
  P_PK_PEDIDOS       integer,  
  P_FK_TIPO_PEDIDOS_NEW integer)  
returns (  
  R_PK_PEDIDOS integer  
)  
as  
  declare variable FkTipoDocumentos smallint;  
begin  
  R_PK_PEDIDOS = 0;  
  select FK_TIPO_DOCUMENTOS from TIPO_PEDIDOS  
  where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS_NEW  
  into :FkTipoDocumentos;  
  if ((FkTipoDocumentos is not null) and (FkTipoDocumentos > 0)) then  
  begin  
    select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO  
    where FK_EMPRESAS      = :P_FK_EMPRESAS  
    and FK_TIPO_DOCUMENTOS = :FkTipoDocumentos  
    into :R_PK_PEDIDOS;  
    if (R_PK_PEDIDOS is null) then  
    begin  
      insert into TIPO_DOCUMENTOS_NUMERACAO  
      (FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)  
      values  
      (:P_FK_EMPRESAS, :FkTipoDocumentos, 0);  
      R_PK_PEDIDOS = 0;  
    end  
    R_PK_PEDIDOS = R_PK_PEDIDOS + 1;  
    update TIPO_DOCUMENTOS_NUMERACAO set  
      NUM_DOC      = :R_PK_PEDIDOS  
    where FK_EMPRESAS      = :P_FK_EMPRESAS  
    and FK_TIPO_DOCUMENTOS = :FkTipoDocumentos;  
  end  
  if (R_PK_PEDIDOS > 0) then  
  begin  
    update PEDIDOS set  
      PK_PEDIDOS      = :R_PK_PEDIDOS,  
      FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS_NEW  
    where FK_EMPRESAS = :P_FK_EMPRESAS  
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS  
    and PK_PEDIDOS    = :P_PK_PEDIDOS;  
  end  
  suspend;  
end ^
```

```
set term ^;
```


1.4.10.4 List of diagrams containing the procedure STP_PROMOVE_PEDIDO

<i>Name</i>	<i>Code</i>
SysPed	SYSPED

1.4.11 Procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS**1.4.11.1 Card of procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS**

<i>Name</i>	STP_UPDATE_SUPPLIERS_FROM_ITEMS
<i>Code</i>	STP_UPDATE_SUPPLIERS_FROM_ITEMS
<i>Parent</i>	Package 'SysPed'
<i>Type</i>	Procedure

1.4.11.2 Code preview of procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS

set term ^;

```

create procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS (
  P_FK_EMPRESAS    integer,
  P_FK_TIPO_PEDIDOS smallint,
  P_FK_PEDIDOS     integer,
  P_FK_TIPO_MOVIM_ESTQ smallint,
  P_FK_CADASTROS   integer
)
returns (
  R_STATUS smallint
)
as
  declare variable PkTipoStatusPedidos smallint;
  declare variable PkTipoMovimEstq    smallint;
  declare variable PkTipoDocumentos   smallint;
  declare variable PkProdutos          smallint;
  declare variable PkAlmoxarifados     smallint;
  declare variable PkLotacoes          smallint;
  declare variable QtdItem             numeric(9, 4);
begin
  R_STATUS = -1;
  if ((P_FK_EMPRESAS is null) or (P_FK_EMPRESAS = 0) or
      (P_FK_TIPO_PEDIDOS is null) or (P_FK_TIPO_PEDIDOS = 0) or
      (P_FK_PEDIDOS is null) or (P_FK_PEDIDOS = 0)) then
    begin
      suspend;
      exit;
    end
  select FK_TIPO_STATUS_PEDIDOS from PEDIDOS
  where FK_EMPRESAS = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and PK_PEDIDOS = :P_FK_PEDIDOS
  into :PkTipoStatusPedidos;
  if ((PkTipoStatusPedidos is null) and (PkTipoStatusPedidos = 0)) then
    begin
      suspend;
      exit;
    end
  select FK_TIPO_MOVIM_ESTQ from TIPO_STATUS_PEDIDOS

```

```

where PK_TIPO_STATUS_PEDIDOS = :PkTipoStatusPedidos
into :PkTipoMovimEstq;
if (PkTipoMovimEstq <> P_FK_TIPO_MOVIM_ESTQ) then
begin
    suspend;
    exit;
end
select FK_TIPO_DOCUMENTOS from TIPO_PEDIDOS
where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
into :PkTipoDocumentos;
if ((PkTipoDocumentos is null) and (PkTipoDocumentos = 0)) then
begin
    suspend;
    exit;
end
for select FK_PRODUTOS, QTD_ITEM, FK_ALMOXARIFADOS, FK_LOTACOES
    from PEDIDOS_ITENS Pdi
where FK_EMPRESAS    = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and FK_PEDIDOS    = :P_FK_PEDIDOS
    into :PkProdutos, :QtdItem, :PkAlmoxarifados, :PkLotacoes do
begin
    select R_STATUS
    from STP_UPDATE_SUPPLIES(:P_FK_EMPRESAS,
        :PkProdutos,
        :PkTipoMovimEstq,
        :PkTipoDocumentos,
        :PkAlmoxarifados,
        0,
        :PkLotacoes,
        0,
        :P_FK_CADASTROS,
        :P_FK_PEDIDOS,
        :QtdItem)
    into :R_STATUS;
end
suspend;
end ^

```

set term ;^;

grant execute on procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS to public;

1.4.11.3 Text of procedure STP_UPDATE_SUPPLIERS FROM ITEMS

set term ^;

```

create procedure %PROC% (
    P_FK_EMPRESAS    integer,
    P_FK_TIPO_PEDIDOS smallint,
    P_FK_PEDIDOS     integer,
    P_FK_TIPO_MOVIM_ESTQ smallint,
    P_FK_CADASTROS   integer
)
returns (
    R_STATUS smallint
)
as
    declare variable PkTipoStatusPedidos smallint;

```

```
declare variable PkTipoMovimEstq    smallint;
declare variable PkTipoDocumentos  smallint;
declare variable PkProdutos         smallint;
declare variable PkAlmoxarifados    smallint;
declare variable PkLotacoes         smallint;
declare variable QtdItem             numeric(9, 4);
begin
  R_STATUS = -1;
  if ((P_FK_EMPRESAS is null) or (P_FK_EMPRESAS = 0) or
    ( P_FK_TIPO_PEDIDOS is null) or (P_FK_TIPO_PEDIDOS = 0) or
    ( P_FK_PEDIDOS is null) or (P_FK_PEDIDOS = 0)) then
  begin
    suspend;
    exit;
  end
  select FK_TIPO_STATUS_PEDIDOS from PEDIDOS
  where FK_EMPRESAS = :P_FK_EMPRESAS
    and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
    and PK_PEDIDOS = :P_FK_PEDIDOS
  into :PkTipoStatusPedidos;
  if ((PkTipoStatusPedidos is null) and (PkTipoStatusPedidos = 0)) then
  begin
    suspend;
    exit;
  end
  select FK_TIPO_MOVIM_ESTQ from TIPO_STATUS_PEDIDOS
  where PK_TIPO_STATUS_PEDIDOS = :PkTipoStatusPedidos
  into :PkTipoMovimEstq;
  if (PkTipoMovimEstq <> P_FK_TIPO_MOVIM_ESTQ) then
  begin
    suspend;
    exit;
  end
  select FK_TIPO_DOCUMENTOS from TIPO_PEDIDOS
  where PK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
  into :PkTipoDocumentos;
  if ((PkTipoDocumentos is null) and (PkTipoDocumentos = 0)) then
  begin
    suspend;
    exit;
  end
  for select FK_PRODUTOS, QTD_ITEM, FK_ALMOXARIFADOS, FK_LOTACOES
    from PEDIDOS_ITENS Pdi
    where FK_EMPRESAS = :P_FK_EMPRESAS
      and FK_TIPO_PEDIDOS = :P_FK_TIPO_PEDIDOS
      and FK_PEDIDOS = :P_FK_PEDIDOS
    into :PkProdutos, :QtdItem, :PkAlmoxarifados, :PkLotacoes do
  begin
    select R_STATUS
    from STP_UPDATE_SUPPLIES(:P_FK_EMPRESAS,
      :PkProdutos,
      :PkTipoMovimEstq,
      :PkTipoDocumentos,
      :PkAlmoxarifados,
      0,
      :PkLotacoes,
      0,
      :P_FK_CADASTROS,
```

```
        :P_FK_PEDIDOS,  
        :QtdItem)  
    into :R_STATUS;  
end  
suspend;  
end ^  
  
set term ;^
```

1.4.11.4 List of diagrams containing the procedure STP_UPDATE_SUPPLIERS_FROM_ITEMS

<i>Name</i>	<i>Code</i>
SysPed	SYSPED