

# Report

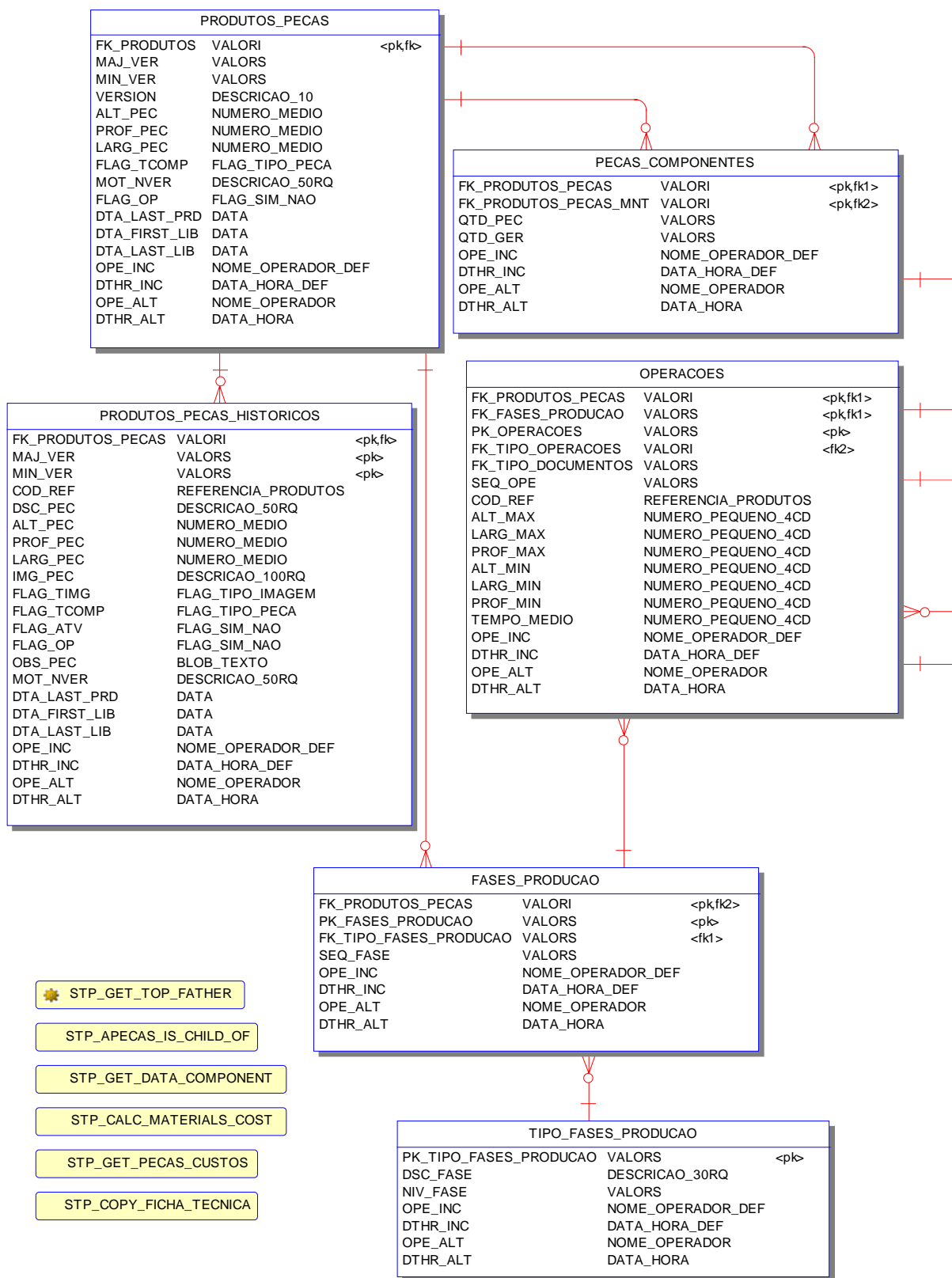
Schleder  
04/04/2006

## Conteúdo

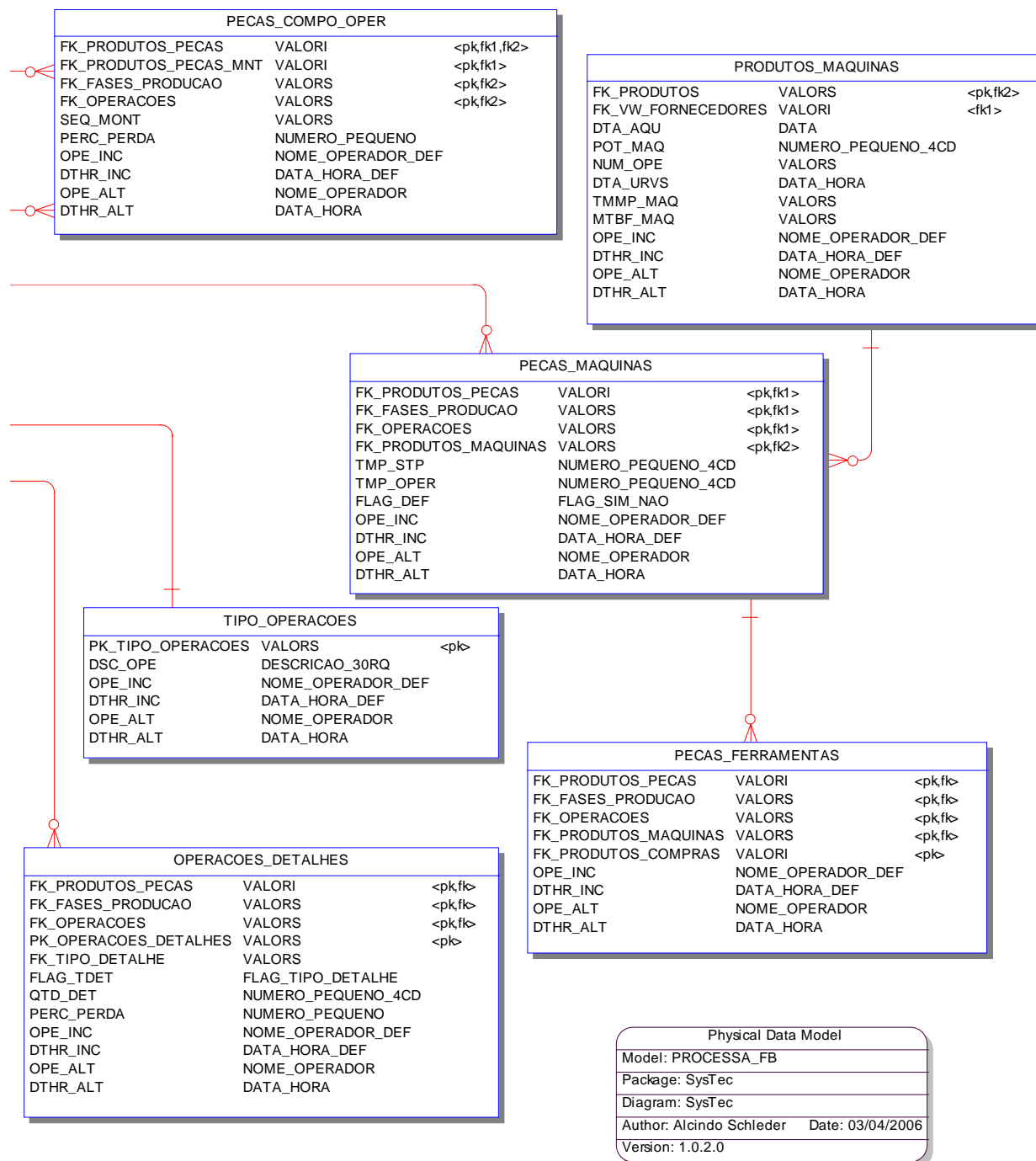
1 Pacote SysTec.....	3
1.1 Diagrama SysTec.....	3
1.1.1 Card of diagram SysTec.....	7
1.2 List of tables.....	7
1.3 Tabelas.....	8
1.3.1 Tabela FASES_PRODUCAO.....	8
1.3.2 Tabela OPERACOES.....	10
1.3.3 Tabela OPERACOES_DETALHES.....	13
1.3.4 Tabela PECAS_COMPONENTES.....	15
1.3.5 Tabela PECAS_COMPO_OPER.....	17
1.3.6 Tabela PECAS_FERRAMENTAS.....	19
1.3.7 Tabela PECAS_MAQUINAS.....	21
1.3.8 Tabela PRODUTOS_MAQUINAS.....	23
1.3.9 Tabela PRODUTOS_PECAS.....	25
1.3.10 Tabela PRODUTOS_PECAS_HISTORICOS.....	28
1.3.11 Tabela TIPO_FASES_PRODUCAO.....	31
1.3.12 Tabela TIPO_OPERACOES.....	33
1.4 Stored Procedures.....	35
1.4.1 Procedure STP_APECAS_IS_CHILD_OF.....	35
1.4.2 Procedure STP_CALC_MATERIALS_COST.....	37
1.4.3 Procedure STP_COPY_FICHA_TECNICA.....	42
1.4.4 Procedure STP_GET_DATA_COMPONENT.....	53
1.4.5 Procedure STP_GET_PECAS_CUSTOS.....	55
1.4.6 Procedure STP_GET_TOP_FATHER.....	57

**1 Pacote SysTec**

**1.1 Diagrama SysTec**



# Módulo de Gerenciamento de Procedimentos Técnicos





**1.1.1 Card of diagram SysTec**

<i>Name</i>	SysTec
<i>Code</i>	SYSTEC
<i>Comment</i>	

**1.2 List of tables**

Name	Comment
TIPO_FASES_PRODUCAO	Tabela que armazena as fases da produção do produto
TIPO_OPERACOES	Tabela que armazena os tipos de operações da ficha técnica (Verbos)
PECAS_FERRAMENTAS	Tabela que armazena as Ferramentas utilizadas nas máquinas
OPERACOES_DETALHES	Tabela que armazena o detalhamento das operações.... podendo ser: Tipo de Acabamento Insumos Serviços
PECAS_MAQUINAS	Tabela que Armazena as máquinas que são usadas para construir determinada peça
PECAS_COMPO_OPER	Tabela que armazena o relacionamento entre Componentes de montagem e operações
PECAS_COMPONENTES	Tabela que armazena o relacionamento e quantidade das peças para um componentes
OPERACOES	Tabela que armazena as operações efetuadas nas peças
PRODUTOS_MAQUINAS	Tabela que armazena as máquinas utilizadas na indústria
FASES_PRODUCAO	Tabela que armazena as fases de produção da Peça
PRODUTOS_PECAS	Tabela que armazena as peças do sistema
PRODUTOS_PECAS_HISTORICOS	Cadastro da ficha técnica das peças com controle de versão.

## 1.3 Tabelas

### 1.3.1 Tabela FASES\_PRODUCAO

#### 1.3.1.1 Card of table FASES\_PRODUCAO

Name	FASES_PRODUCAO
Code	FASES_PRODUCAO
Comment	Tabela que armazena as fases de produção da Peça
Primary Key	PK_FASES_PRODUCAO

#### 1.3.1.2 Lista de Colunas da Tabela FASES\_PRODUCAO

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
PK_FASES_PRODUCAO	SMALLINT	VALORS	
FK_TIPO_FASES_PRODUCAO	SMALLINT	VALORS	Código do Tipo da fase de operação
SEQ_FASE	SMALLINT	VALORS	Sequencia da fase de produção
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 FASES\_PRODUCAO

Name	Code	Class Name
FK_OPERACOES_FASES_PRODUCAO	FK_OPERACOES_FASES_PRODUCAO	Reference
FK_FASES_PRODUCAO_TIPO_FASE_PRO	FK_FASES_PRODUCAO_TIPO_FASE_PRO	Reference
FK_FASES_PRODUCAO_PRODUTOS_PECA	FK_FASES_PRODUCAO_PRODUTOS_PECA	Reference

#### 1.3.1.4 Lista de Chaves da Tabela FASES\_PRODUCAO

Name	Code	Primary
PK_FASES_PRODUCAO	PK_FASES_PRODUCAO	X

#### 1.3.1.5 Chave PK\_FASES\_PRODUCAO da Tabela FASES\_PRODUCAO

##### 1.3.1.5.1 Lista de Colunas da Chave PK\_FASES\_PRODUCAO

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



Name	Code
PK_FASES_PRODUCAO	PK_FASES_PRODUCAO
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS

**1.3.1.5.2 Dependências da Chave PK\_FASES\_PRODUCAO**

Name	Code	Class Name
PK_FASES_PRODUCAO	PK_FASES_PRODUCAO	Index
FK_OPERACOES_FASES_PRODUCA	FK_OPERACOES_FASES_PRODUCA	Reference

**1.3.1.6 Lista de Índices FASES\_PRODUCAO**

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

**1.3.1.7 Índice PK\_FASES\_PRODUCAO da Tabela FASES\_PRODUCAO****1.3.1.7.1 Lista de Colunas do Índice PK\_FASES\_PRODUCAO**

Name	Code	Sort
PK_FASES_PRODUCAO	PK_FASES_PRODUCAO	Ascending
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending

**1.3.1.8 Lista de Triggers da Tabela FASES\_PRODUCAO**

Name	Code
TBIG_FASES_PRODUCAO	TBIG_FASES_PRODUCAO
TBU0_FASES_PRODUCAO	TBU0_FASES_PRODUCAO

## 1.3.2 Tabela OPERACOES

## 1.3.2.1 Card of table OPERACOES

Name	OPERACOES
Code	OPERACOES
Comment	Tabela que armazena as operações efetuadas nas peças
Primary Key	PK_OPERACOES

## 1.3.2.2 Lista de Colunas da Tabela OPERACOES

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase de produção
PK_OPERACOES	SMALLINT	VALORS	Código da Operação
FK_TIPO_OPERACOES	INTEGER	VALORI	Código do Tipo da Operação
FK_TIPO_DOCUMENTOS	SMALLINT	VALORS	Código do tipo de documento que pode gerar esta operação
SEQ_OPE	SMALLINT	VALORS	Número de sequencia da operação
COD_REF	VARCHAR(20)	REFERENCIA_PRODUTOS	Código de referência da Peça
ALT_MAX	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Altura Máxima da Operação
LARG_MAX	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Largura Máxima da Operação
PROF_MAX	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Profundidade Máxima da Operação
ALT_MIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Altura Mínima da Operação
LARG_MIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Largura Mínima da Operação
PROF_MIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Profundidade Mínima da Operação
TEMPO_MEDIO	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
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

## 1.3.2.3 Dependências da Tabela OPERACOES

Name	Code	Class Name
FK_PECAS_MAUQUINA_OPERACOES	FK_PECAS_MAUQUINA_OPERACOES	Reference
FK_PECAS_COMPO_O_OPERACOES	FK_PECAS_COMPO_O_OPERACOES	Reference
FK_OPERACOES_DET_OPERACOES	FK_OPERACOES_DET_OPERACOES	Reference

Name	Code	Class Name
FK_ORDENS_OPERAC_OPERACOE S	FK_ORDENS_OPERAC_OPERACOE S	Reference
FK_OPERACOES_FASES_PRODU CA	FK_OPERACOES_FASES_PRODU CA	Reference
FK_OPERACOES_TIPO_OPERAC AO	FK_OPERACOES_TIPO_OPERAC AO	Reference
OPERACOES	OPERACOES	Shortcut of table

#### 1.3.2.4 Lista de Chaves da Tabela OPERACOES

Name	Code	Primary
PK_OPERACOES	PK_OPERACOES	X

#### 1.3.2.5 Chave PK\_OPERACOES da Tabela OPERACOES

##### 1.3.2.5.1 Lista de Colunas da Chave PK\_OPERACOES

Name	Code
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
PK_OPERACOES	PK_OPERACOES

##### 1.3.2.5.2 Dependências da Chave PK\_OPERACOES

Name	Code	Class Name
PK_OPERACOES	PK_OPERACOES	Index
FK_ORDENS_OPERAC_OPERACOE S	FK_ORDENS_OPERAC_OPERACOE S	Reference
FK_PECAS_MAQUINA_OPERACO ES	FK_PECAS_MAQUINA_OPERACO ES	Reference
FK_PECAS_COMPO_O_OPERACO ES	FK_PECAS_COMPO_O_OPERACO ES	Reference
FK_OPERACOES_DET_OPERACO ES	FK_OPERACOES_DET_OPERACO ES	Reference

#### 1.3.2.6 Lista de Índices OPERACOES

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_OPERACO ES	PK_OPERACO ES	X		X		

#### 1.3.2.7 Índice PK\_OPERACOES da Tabela OPERACOES

##### 1.3.2.7.1 Lista de Colunas do Índice PK\_OPERACOES

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

Name	Code	Sort
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO	Ascending
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
PK_OPERACOES	PK_OPERACOES	Ascending

**1.3.2.8 Lista de Triggers da Tabela OPERACOES**

Name	Code
TBIG_OPERACOES	TBIG_OPERACOES
TBU0_OPERACOES	TBU0_OPERACOES

## 1.3.3 Tabela OPERACOES\_DETALHES

## 1.3.3.1 Card of table OPERACOES\_DETALHES

Name	OPERACOES_DETALHES
Code	OPERACOES_DETALHES
Comment	Tabela que armazena o detalhamento das operações.... podendo ser: Tipo de Acabamento Insumos Serviços
Primary Key	PK_OPERACOES_DETALHES

## 1.3.3.2 Lista de Colunas da Tabela OPERACOES\_DETALHES

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase de produção
FK_OPERACOES	SMALLINT	VALORS	Código da Operação
PK_OPERACOES_DETALHES	SMALLINT	VALORS	Sequência do detalhamento
FK_TIPO_DETALHE	SMALLINT	VALORS	Código do Detalhe pode ser: Tipo de Acabamento Produtos (que pode ser serviços, produtos, pecas, etc..)
FLAG_TDET	SMALLINT	FLAG_TIPO_DETALHE	Flag que indica o tipo de Detalhe usado na operação 0 ==> Produtos 1 ==> Serviços 2 ==> Tipo de Acabamento
QTD_DET	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Quantidade do detalhe aplicado na operação
PERC_PERDA	NUMERIC(9,2)	NUMERO_PEQUENO	percentual de perda na confecção da peça
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 OPERACOES\_DETALHES

Name	Code	Class Name
FK_OPERACOES_DET_OPERACOES	FK_OPERACOES_DET_OPERACOES	Reference

## 1.3.3.4 Lista de Chaves da Tabela OPERACOES\_DETALHES

Name	Code	Primary
PK_OPERACOES_DETALHES	PK_OPERACOES_DETALHES	X

## 1.3.3.5 Chave PK\_OPERACOES\_DETALHES da Tabela OPERACOES\_DETALHES

## 1.3.3.5.1 Lista de Colunas da Chave PK\_OPERACOES\_DETALHES

Name	Code
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
FK_OPERACOES	FK_OPERACOES
PK_OPERACOES_DETALHES	PK_OPERACOES_DETALHES

## 1.3.3.5.2 Dependências da Chave PK\_OPERACOES\_DETALHES

Name	Code	Class Name
PK_OPERACOES_DETALHES	PK_OPERACOES_DETALHES	Index

## 1.3.3.6 Lista de Índices OPERACOES\_DETALHES

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

## 1.3.3.7 Índice PK\_OPERACOES\_DETALHES da Tabela OPERACOES\_DETALHES

## 1.3.3.7.1 Lista de Colunas do Índice PK\_OPERACOES\_DETALHES

Name	Code	Sort
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO	Ascending
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
FK_OPERACOES	FK_OPERACOES	Ascending
PK_OPERACOES_DETALHES	PK_OPERACOES_DETALHES	Ascending

## 1.3.3.8 Lista de Triggers da Tabela OPERACOES\_DETALHES

Name	Code
TBIG_OPERACOES_DETALHES	TBIG_OPERACOES_DETALHES
TBU0_OPERACOES_DETALHES	TBU0_OPERACOES_DETALHES

## 1.3.4 Tabela PECAS\_COMPONENTES

## 1.3.4.1 Card of table PECAS\_COMPONENTES

Name	PECAS_COMPONENTES
Code	PECAS_COMPONENTES
Comment	Tabela que armazena o relacionamento e quantidade das peças para um componentes
Primary Key	PK_PECAS_COMPONENTES

## 1.3.4.2 Lista de Colunas da Tabela PECAS\_COMPONENTES

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_PRODUTOS_PECAS_MNT	INTEGER	VALORI	Código da Peça para Montagem
QTD_PEC	SMALLINT	VALORS	Quantidade de peças usadas no componente
QTD_GER	SMALLINT	VALORS	Quantidade de peças que irá gerar utilizando a peça filha
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

## 1.3.4.3 Dependências da Tabela PECAS\_COMPONENTES

Name	Code	Class Name
FK_PECAS_COMPO_O_PECAS_COMPONE	FK_PECAS_COMPO_O_PECAS_COMPONE	Reference
FK_PECAS_COMPONE_PRODUTOS_PECAS	FK_PECAS_COMPONE_PRODUTOS_PECAS	Reference
FK_PECAS_COMPONENT_PRODUTOS_PEC	FK_PECAS_COMPONENT_PRODUTOS_PEC	Reference

## 1.3.4.4 Lista de Chaves da Tabela PECAS\_COMPONENTES

Name	Code	Primary
PK_PECAS_COMPONENTES	PK_PECAS_COMPONENTES	X

## 1.3.4.5 Chave PK\_PECAS\_COMPONENTES da Tabela PECAS\_COMPONENTES

## 1.3.4.5.1 Lista de Colunas da Chave PK\_PECAS\_COMPONENTES

Name	Code
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
FK_PRODUTOS_PECAS_MNT	FK_PRODUTOS_PECAS_MNT

## 1.3.4.5.2 Dependências da Chave PK\_PECAS\_COMPONENTES

Name	Code	Class Name
PK_PECAS_COMPONENTES	PK_PECAS_COMPONENTES	Index
FK_PECAS_COMPO_O_PECAS_C OMPONE	FK_PECAS_COMPO_O_PECAS_C OMPONE	Reference

## 1.3.4.6 Lista de Índices PECAS\_COMPONENTES

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_PECAS_C OMPONENTES	PK_PECAS_C OMPONENTES	X		X		

## 1.3.4.7 Índice PK\_PECAS\_COMPONENTES da Tabela PECAS\_COMPONENTES

## 1.3.4.7.1 Lista de Colunas do Índice PK\_PECAS\_COMPONENTES

Name	Code	Sort
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
FK_PRODUTOS_PECAS_MNT	FK_PRODUTOS_PECAS_MNT	Ascending

## 1.3.4.8 Lista de Triggers da Tabela PECAS\_COMPONENTES

Name	Code
TBU0_PECAS_COMPONENTES	TBU0_PECAS_COMPONENTES



## 1.3.5 Tabela PECAS\_COMPO\_OPER

## 1.3.5.1 Card of table PECAS\_COMPO\_OPER

Name	PECAS_COMPO_OPER
Code	PECAS_COMPO_OPER
Comment	Tabela que armazena o relacionamento entre Componentes de montagem e operações
Primary Key	PK_PECAS_COMPO_OPER

## 1.3.5.2 Lista de Colunas da Tabela PECAS\_COMPO\_OPER

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_PRODUTOS_PECAS_MNT	INTEGER	VALORI	Código da Peça para Montagem
FK_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase de produção
FK_OPERACOES	SMALLINT	VALORS	Código da Operação
SEQ_MONT	SMALLINT	VALORS	Sequência de montagem
PERC_PERDA	NUMERIC(9,2)	NUMERO_PEQUENO	percentual de perda na confecção da peça
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

## 1.3.5.3 Dependências da Tabela PECAS\_COMPO\_OPER

Name	Code	Class Name
FK_PECAS_COMPO_O_PECAS_COMPONE	FK_PECAS_COMPO_O_PECAS_COMPONE	Reference
FK_PECAS_COMPO_O_OPERACOES	FK_PECAS_COMPO_O_OPERACOES	Reference

## 1.3.5.4 Lista de Chaves da Tabela PECAS\_COMPO\_OPER

Name	Code	Primary
PK_PECAS_COMPO_OPER	PK_PECAS_COMPO_OPER	X

## 1.3.5.5 Chave PK\_PECAS\_COMPO\_OPER da Tabela PECAS\_COMPO\_OPER

## 1.3.5.5.1 Lista de Colunas da Chave PK\_PECAS\_COMPO\_OPER

Name	Code
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS

Name	Code
FK_PRODUTOS_PECAS_MNT	FK_PRODUTOS_PECAS_MNT
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO
FK_OPERACOES	FK_OPERACOES

#### 1.3.5.5.2 Dependências da Chave PK\_PECAS\_COMPO\_OPER

Name	Code	Class Name
PK_PECAS_COMPO_OPER	PK_PECAS_COMPO_OPER	Index

#### 1.3.5.6 Lista de Índices PECAS\_COMPO\_OPER

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

#### 1.3.5.7 Índice PK\_PECAS\_COMPO\_OPER da Tabela PECAS\_COMPO\_OPER

##### 1.3.5.7.1 Lista de Colunas do Índice PK\_PECAS\_COMPO\_OPER

Name	Code	Sort
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
FK_PRODUTOS_PECAS_MNT	FK_PRODUTOS_PECAS_MNT	Ascending
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO	Ascending
FK_OPERACOES	FK_OPERACOES	Ascending

#### 1.3.5.8 Lista de Triggers da Tabela PECAS\_COMPO\_OPER

Name	Code
TBU0_PECAS_COMPO_OPER	TBU0_PECAS_COMPO_OPER

## 1.3.6 Tabela PECAS\_FERRAMENTAS

## 1.3.6.1 Card of table PECAS\_FERRAMENTAS

Name	PECAS_FERRAMENTAS
Code	PECAS_FERRAMENTAS
Comment	Tabela que armazena as Ferramentas utilizadas nas máquinas
Primary Key	PK_PECAS_FERRAMENTAS

## 1.3.6.2 Lista de Colunas da Tabela PECAS\_FERRAMENTAS

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase de produção
FK_OPERACOES	SMALLINT	VALORS	Código da Operação
FK_PRODUTOS_MAQUINAS	SMALLINT	VALORS	Código da Máquina
FK_PRODUTOS_COMPRAS	INTEGER	VALORI	Código do Produto que representa as ferramentas utilizadas na máquina
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

## 1.3.6.3 Dependências da Tabela PECAS\_FERRAMENTAS

Name	Code	Class Name
FK_PECAS_FERRAME_PECAS_MAQUINA	FK_PECAS_FERRAME_PECAS_MAQUINA	Reference

## 1.3.6.4 Lista de Chaves da Tabela PECAS\_FERRAMENTAS

Name	Code	Primary
PK_PECAS_FERRAMENTAS	PK_PECAS_FERRAMENTAS	X

## 1.3.6.5 Chave PK\_PECAS\_FERRAMENTAS da Tabela PECAS\_FERRAMENTAS

## 1.3.6.5.1 Lista de Colunas da Chave PK\_PECAS\_FERRAMENTAS

Name	Code
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
FK_OPERACOES	FK_OPERACOES

Name	Code
FK_PRODUTOS_MAQUINAS	FK_PRODUTOS_MAQUINAS
FK_PRODUTOS_COMPRAS	FK_PRODUTOS_COMPRAS

**1.3.6.5.2 Dependências da Chave PK\_PECAS\_FERRAMENTAS**

Name	Code	Class Name
PK_PECAS_FERRAMENTAS	PK_PECAS_FERRAMENTAS	Index

**1.3.6.6 Lista de Índices PECAS\_FERRAMENTAS**

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

**1.3.6.7 Índice PK\_PECAS\_FERRAMENTAS da Tabela PECAS\_FERRAMENTAS****1.3.6.7.1 Lista de Colunas do Índice PK\_PECAS\_FERRAMENTAS**

Name	Code	Sort
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO	Ascending
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
FK_OPERACOES	FK_OPERACOES	Ascending
FK_PRODUTOS_MAQUINAS	FK_PRODUTOS_MAQUINAS	Ascending
FK_PRODUTOS_COMPRAS	FK_PRODUTOS_COMPRAS	Ascending

**1.3.6.8 Lista de Triggers da Tabela PECAS\_FERRAMENTAS**

Name	Code
TBU0_PECAS_FERRAMENTAS	TBU0_PECAS_FERRAMENTAS

## 1.3.7 Tabela PECAS\_MAQUINAS

## 1.3.7.1 Card of table PECAS\_MAQUINAS

Name	PECAS_MAQUINAS
Code	PECAS_MAQUINAS
Comment	Tabela que Armazena as máquinas que são usadas para construir determinada peça
Primary Key	PK_PECAS_MAQUINAS

## 1.3.7.2 Lista de Colunas da Tabela PECAS\_MAQUINAS

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código da Peça
FK_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase de produção
FK_OPERACOES	SMALLINT	VALORS	Código da Operação
FK_PRODUTOS_MAQUINAS	SMALLINT	VALORS	Código da Máquina
TMP_STP	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Tempo médio de Setup da Peça na máquina
TMP_OPER	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Tempo médio de operação da peça na máquina
FLAG_DEF	SMALLINT	FLAG_SIM_NAO	Flag que indica que a máquina é a operação default para o cálculo de custos da peça
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

## 1.3.7.3 Dependências da Tabela PECAS\_MAQUINAS

Name	Code	Class Name
FK_PECAS_FERRAME_PECAS_MAQUINA	FK_PECAS_FERRAME_PECAS_MAQUINA	Reference
FK_PECAS_MAQUINA_OPERACOES	FK_PECAS_MAQUINA_OPERACOES	Reference
FK_PECAS_MAQUINA_MAQUINAS	FK_PECAS_MAQUINA_MAQUINAS	Reference

## 1.3.7.4 Lista de Chaves da Tabela PECAS\_MAQUINAS

Name	Code	Primary
PK_PECAS_MAQUINAS	PK_PECAS_MAQUINAS	X

## 1.3.7.5 Chave PK\_PECAS\_MAQUINAS da Tabela PECAS\_MAQUINAS

## 1.3.7.5.1 Lista de Colunas da Chave PK\_PECAS\_MAQUINAS

Name	Code
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
FK_OPERACOES	FK_OPERACOES
FK_PRODUTOS_MAQUINAS	FK_PRODUTOS_MAQUINAS

## 1.3.7.5.2 Dependências da Chave PK\_PECAS\_MAQUINAS

Name	Code	Class Name
PK_PECAS_MAQUINAS	PK_PECAS_MAQUINAS	Index
FK_PECAS_FERRAME_PECAS_M AQUINA	FK_PECAS_FERRAME_PECAS_M AQUINA	Reference

## 1.3.7.6 Lista de Índices PECAS\_MAQUINAS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_PECAS_M AQUINAS	PK_PECAS_M AQUINAS	X		X		

## 1.3.7.7 Índice PK\_PECAS\_MAQUINAS da Tabela PECAS\_MAQUINAS

## 1.3.7.7.1 Lista de Colunas do Índice PK\_PECAS\_MAQUINAS

Name	Code	Sort
FK_FASES_PRODUCAO	FK_FASES_PRODUCAO	Ascending
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
FK_OPERACOES	FK_OPERACOES	Ascending
FK_PRODUTOS_MAQUINAS	FK_PRODUTOS_MAQUINAS	Ascending

## 1.3.7.8 Lista de Triggers da Tabela PECAS\_MAQUINAS

Name	Code
TBU0_PECAS_MAQUINAS	TBU0_PECAS_MAQUINAS

## 1.3.8 Tabela PRODUTOS\_MAQUINAS

## 1.3.8.1 Card of table PRODUTOS\_MAQUINAS

Name	PRODUTOS_MAQUINAS
Code	PRODUTOS_MAQUINAS
Comment	Tabela que armazena as máquinas utilizadas na indústria
Primary Key	PK_PRODUTOS_MAQUINAS

## 1.3.8.2 Lista de Colunas da Tabela PRODUTOS\_MAQUINAS

Name	Data Type	Domain	Comment
FK_PRODUTOS	SMALLINT	VALORS	Código da Máquina
FK_VW_FORNecedores	INTEGER	VALORI	Código do Fornecedor
DTA_AQU	DATE	DATA	Data de Aquisição
POT_MAQ	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Potência da máquina
NUM_OPE	SMALLINT	VALORS	Quantidade de Operadores na máquina
DTA_URVS	TIMESTAMP	DATA_HORA	Data da Última Revisão
TMMP_MAQ	SMALLINT	VALORS	Tempo médio de manutenção preventiva em dias
MTBF_MAQ	SMALLINT	VALORS	Tempo médio entre defeitos da máquina
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

## 1.3.8.3 Dependências da Tabela PRODUTOS\_MAQUINAS

Name	Code	Class Name
FK_PECAS_MAQUINA_MAQUINAS	FK_PECAS_MAQUINA_MAQUINAS	Reference
FK_ORDENS_ALOCAC_MAQUINAS	FK_ORDENS_ALOCAC_MAQUINAS	Reference
FK_MAQUINAS_CADASTROS	FK_MAQUINAS_CADASTROS	Reference
FK_PRODUTOS_MAQUINAS_PRODUTOS	FK_PRODUTOS_MAQUINAS_PRODUTOS	Reference
PRODUTOS_MAQUINAS	PRODUTOS_MAQUINAS	Shortcut of table

## 1.3.8.4 Lista de Chaves da Tabela PRODUTOS\_MAQUINAS

Name	Code	Primary
PK_PRODUTOS_MAQUINAS	PK_PRODUTOS_MAQUINAS	X

**1.3.8.5 Chave PK\_PRODUTOS\_MAQUINAS da Tabela PRODUTOS\_MAQUINAS****1.3.8.5.1 Lista de Colunas da Chave PK\_PRODUTOS\_MAQUINAS**

Name	Code
FK_PRODUTOS	FK_PRODUTOS

**1.3.8.5.2 Dependências da Chave PK\_PRODUTOS\_MAQUINAS**

Name	Code	Class Name
PK_PRODUTOS_MAQUINAS	PK_PRODUTOS_MAQUINAS	Index
FK_ORDENS_ALOCAC_MAQUINAS	FK_ORDENS_ALOCAC_MAQUINAS	Reference
FK_PECAS_MAQUINA_MAQUINAS	FK_PECAS_MAQUINA_MAQUINAS	Reference

**1.3.8.6 Lista de Índices PRODUTOS\_MAQUINAS**

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

**1.3.8.7 Índice PK\_PRODUTOS\_MAQUINAS da Tabela PRODUTOS\_MAQUINAS****1.3.8.7.1 Lista de Colunas do Índice PK\_PRODUTOS\_MAQUINAS**

Name	Code	Sort
FK_PRODUTOS	FK_PRODUTOS	Ascending

**1.3.8.8 Lista de Triggers da Tabela PRODUTOS\_MAQUINAS**

Name	Code
TBU0_PRODUTOS_MAQUINAS	TBU0_PRODUTOS_MAQUINAS



## 1.3.9 Tabela PRODUTOS\_PECAS

## 1.3.9.1 Card of table PRODUTOS\_PECAS

Name	PRODUTOS_PECAS
Code	PRODUTOS_PECAS
Comment	Tabela que armazena as peças do sistema
Primary Key	PK_PRODUTOS_PECAS

## 1.3.9.2 Lista de Colunas da Tabela PRODUTOS\_PECAS

Name	Data Type	Domain	Comment
FK_PRODUTOS	INTEGER	VALORI	Código da Produto
MAJ_VER	SMALLINT	VALORS	Número maior da Versão da Peça
MIN_VER	SMALLINT	VALORS	Número menor do número da versão
VERSION	VARCHAR(10)	DESCRICAO_10	Valor concatenado indicando a versão da peça
ALT_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Altura da Peça
PROF_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Profundidade da Peça
LARG_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Largura da Peça
FLAG_TCOMPM	SMALLINT	FLAG_TIPO_Peca	Flag que indica o tipo de peça que está sendo cadastrada 0 ==> Peça 1 ==> Componente 2 ==> Produto Semi-Acabado 3 ==> Produto Lista 4 ==> Produto Acabados
MOT_NVER	VARCHAR(50)	DESCRICAO_50RQ	Motivo para a criação da nova versão
FLAG_OP	SMALLINT	FLAG_SIM_NAO	Flag que indica se existe alguma Ordem de Produção que está em andamento com esta ficha técnica
DTA_LAST_PRD	DATE	DATA	Data da última ordem de produção da ficha técnica
DTA_FIRST_LIB	DATE	DATA	Data da primeira liberação da ficha técnica para a produção
DTA_LAST_LIB	DATE	DATA	Data da última liberação da ficha técnica para a produção
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

## 1.3.9.3 Dependências da Tabela PRODUTOS\_PECAS

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

Name	Code	Class Name
FK_PRODUTOS_PECAS_PRODUTOS_PHST	FK_PRODUTOS_PECAS_PRODUTOS_PHST	Reference
FK_PECAS_COMPONE_PRODUTOS_PECAS	FK_PECAS_COMPONE_PRODUTOS_PECAS	Reference
FK_PECAS_COMPONENT_PRODUTOS_PECAS	FK_PECAS_COMPONENT_PRODUTOS_PECAS	Reference
FK_FASES_PRODUCA_PRODUTOS_PECAS	FK_FASES_PRODUCA_PRODUTOS_PECAS	Reference
FK_PRODUTOS_PECAS_PRODUTOS	FK_PRODUTOS_PECAS_PRODUTOS	Reference

#### 1.3.9.4 Lista de Chaves da Tabela PRODUTOS\_PECAS

Name	Code	Primary
PK_PRODUTOS_PECAS	PK_PRODUTOS_PECAS	X

#### 1.3.9.5 Chave PK\_PRODUTOS\_PECAS da Tabela PRODUTOS\_PECAS

##### 1.3.9.5.1 Lista de Colunas da Chave PK\_PRODUTOS\_PECAS

Name	Code
FK_PRODUTOS	FK_PRODUTOS

##### 1.3.9.5.2 Dependências da Chave PK\_PRODUTOS\_PECAS

Name	Code	Class Name
PK_PRODUTOS_PECAS	PK_PRODUTOS_PECAS	Index
FK_PRODUTOS_PECAS_PRODUTOS_PHST	FK_PRODUTOS_PECAS_PRODUTOS_PHST	Reference
FK_PECAS_COMPONE_PRODUTOS_PECAS	FK_PECAS_COMPONE_PRODUTOS_PECAS	Reference
FK_PECAS_COMPONENT_PRODUTOS_PECAS	FK_PECAS_COMPONENT_PRODUTOS_PECAS	Reference
FK_FASES_PRODUCA_PRODUTOS_PECAS	FK_FASES_PRODUCA_PRODUTOS_PECAS	Reference

#### 1.3.9.6 Lista de Índices PRODUTOS\_PECAS

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

#### 1.3.9.7 Índice PK\_PRODUTOS\_PECAS da Tabela PRODUTOS\_PECAS

##### 1.3.9.7.1 Lista de Colunas do Índice PK\_PRODUTOS\_PECAS

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

Name	Code	Sort
FK_PRODUTOS	FK_PRODUTOS	Ascending

**1.3.9.8 Lista de Triggers da Tabela PRODUTOS\_PECAS**

Name	Code
TBU0_PRODUTOS_PECAS	TBU0_PRODUTOS_PECAS

## 1.3.10 Tabela PRODUTOS\_PECAS\_HISTORICOS

## 1.3.10.1 Card of table PRODUTOS\_PECAS\_HISTORICOS

Name	PRODUTOS_PECAS_HISTORICOS
Code	PRODUTOS_PECAS_HISTORICOS
Comment	Cadastro da ficha técnica das peças com controle de versão.
Primary Key	PK_PRODUTOS_PECAS_HISTORICOS

## 1.3.10.2 Lista de Colunas da Tabela PRODUTOS\_PECAS\_HISTORICOS

Name	Data Type	Domain	Comment
FK_PRODUTOS_PECAS	INTEGER	VALORI	Código do Produto
MAJ_VER	SMALLINT	VALORS	Número maior da Versão da Peça
MIN_VER	SMALLINT	VALORS	Número menor do número da versão
COD_REF	VARCHAR(20)	REFERENCIA_PRODUTOS	Código de Referência
DSC_PEC	VARCHAR(50)	DESCRICAO_50RQ	Descrição da Peça
ALT_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Altura da Peça
PROF_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Profundidade da Peça
LARG_PEC	NUMERIC(11,2)	NUMERO_MEDIO	Largura da Peça
IMG_PEC	VARCHAR(100)	DESCRICAO_100RQ	endereço o disco da imagem da peça
FLAG_TIMG	SMALLINT	FLAG_TIPO_IMAGEM	Flag que indica o tipo da imagem 0 ==> Jpg 1 ==> Wmf 2 ==> Bmp
FLAG_TCO MP	SMALLINT	FLAG_TIPO_Peca	Flag que indica o tipo de peça que está sendo cadastrada 0 ==> Peça 1 ==> Componente 2 ==> Produto Semi-Acabado 3 ==> Produto Lista 4 ==> Produto Acabados
FLAG_ATV	SMALLINT	FLAG_SIM_NAO	Flag que indica o status da ficha técnica. A primeira inserção a ficha técnica nasce bloqueada (não ativa) para gerar OP's e versões novas...
FLAG_OP	SMALLINT	FLAG_SIM_NAO	Flag que indica se existe alguma Ordem de Produção que está em andamento com esta ficha técnica
OBS_PEC	BLOB SUB_TYPE 1 SEGMENT SIZE 80	BLOB_TEXTO	Observações da Peça
MOT_NVER	VARCHAR(50)	DESCRICAO_50RQ	Motivo para a criação da nova versão

Name	Data Type	Domain	Comment
DTA_LAST_PRD	DATE	DATA	Data da última ordem de produção da ficha técnica
DTA_FIRST_LIB	DATE	DATA	Data da primeira liberação da ficha técnica para a produção
DTA_LAST_LIB	DATE	DATA	Data da última liberação da ficha técnica para a produção
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

### 1.3.10.3 Dependências da Tabela PRODUTOS\_PECAS\_HISTORICOS

Name	Code	Class Name
FK_PRODUTOS_PECAS_PRODUTOS_PHST	FK_PRODUTOS_PECAS_PRODUTOS_PHST	Reference

### 1.3.10.4 Lista de Chaves da Tabela PRODUTOS\_PECAS\_HISTORICOS

Name	Code	Primary
PK_PRODUTOS_PECAS_HISTORICOS	PK_PRODUTOS_PECAS_HISTORICOS	X

### 1.3.10.5 Chave PK\_PRODUTOS\_PECAS\_HISTORICOS da Tabela PRODUTOS\_PECAS\_HISTORICOS

#### 1.3.10.5.1 Lista de Colunas da Chave PK\_PRODUTOS\_PECAS\_HISTORICOS

Name	Code
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS
MAJ_VER	MAJ_VER
MIN_VER	MIN_VER

#### 1.3.10.5.2 Dependências da Chave PK\_PRODUTOS\_PECAS\_HISTORICOS

Name	Code	Class Name
PK_PRODUTOS_PECAS_HISTORICOS	PK_PRODUTOS_PECAS_HISTORICOS	Index

### 1.3.10.6 Lista de Índices PRODUTOS\_PECAS\_HISTORICOS

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

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
IDX_FICHA_TECNICA_01	IDX_FICHA_TECNICA_01					

### 1.3.10.7 Índice PK\_PRODUTOS\_PECAS\_HISTORICOS da Tabela PRODUTOS\_PECAS\_HISTORICOS

#### 1.3.10.7.1 Lista de Colunas do Índice PK\_PRODUTOS\_PECAS\_HISTORICOS

Name	Code	Sort
FK_PRODUTOS_PECAS	FK_PRODUTOS_PECAS	Ascending
MAJ_VER	MAJ_VER	Ascending
MIN_VER	MIN_VER	Ascending

### 1.3.10.8 Índice IDX\_FICHA\_TECNICA\_01 da Tabela PRODUTOS\_PECAS\_HISTORICOS

#### 1.3.10.8.1 Lista de Colunas do Índice IDX\_FICHA\_TECNICA\_01

Name	Code	Sort
COD_REF	COD_REF	Ascending
MAJ_VER	MAJ_VER	Ascending
MIN_VER	MIN_VER	Ascending

### 1.3.10.9 Lista de Triggers da Tabela PRODUTOS\_PECAS\_HISTORICOS

Name	Code
TBU0_PRODUTOS_PECAS_HISTORICOS	TBU0_PRODUTOS_PECAS_HISTORICOS

## 1.3.11 Tabela TIPO\_FASES\_PRODUCAO

## 1.3.11.1 Card of table TIPO\_FASES\_PRODUCAO

Name	TIPO_FASES_PRODUCAO
Code	TIPO_FASES_PRODUCAO
Comment	Tabela que armazena as fases da produção do produto
Primary Key	PK_TIPO_FASES_PRODUCAO

## 1.3.11.2 Lista de Colunas da Tabela TIPO\_FASES\_PRODUCAO

Name	Data Type	Domain	Comment
PK_TIPO_FASES_PRODUCAO	SMALLINT	VALORS	Código da Fase da Produção
DSC_FASE	VARCHAR(30)	DESCRICAO_30RQ	Descrição da Fase
NIV_FASE	SMALLINT	VALORS	Nível da fase da produção
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

## 1.3.11.3 Dependências da Tabela TIPO\_FASES\_PRODUCAO

Name	Code	Class Name
FK_FASES_PRODUCAO_TIPO_FASE_PRO	FK_FASES_PRODUCAO_TIPO_FASE_PRO	Reference

## 1.3.11.4 Lista de Chaves da Tabela TIPO\_FASES\_PRODUCAO

Name	Code	Primary
PK_TIPO_FASES_PRODUCAO	PK_TIPO_FASES_PRODUCAO	X

## 1.3.11.5 Chave PK\_TIPO\_FASES\_PRODUCAO da Tabela TIPO\_FASES\_PRODUCAO

## 1.3.11.5.1 Lista de Colunas da Chave PK\_TIPO\_FASES\_PRODUCAO

Name	Code
PK_TIPO_FASES_PRODUCAO	PK_TIPO_FASES_PRODUCAO

## 1.3.11.5.2 Dependências da Chave PK\_TIPO\_FASES\_PRODUCAO

Name	Code	Class Name
PK_TIPO_FASES_PRODUCAO	PK_TIPO_FASES_PRODUCAO	Index

## 1.3.11.6 Lista de Índices TIPO\_FASES\_PRODUCAO

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
------	------	--------	---------	---------	-------------	---------------

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

### 1.3.11.7 Índice PK\_TIPO\_FASES\_PRODUCAO da Tabela TIPO\_FASES\_PRODUCAO

#### 1.3.11.7.1 Lista de Colunas do Índice PK\_TIPO\_FASES\_PRODUCAO

Name	Code	Sort
PK_TIPO_FASES_PRODUCAO	PK_TIPO_FASES_PRODUCAO	Ascending

### 1.3.11.8 Lista de Triggers da Tabela TIPO\_FASES\_PRODUCAO

Name	Code
TBIG_TIPO_FASES_PRODUCAO	TBIG_TIPO_FASES_PRODUCAO
TBU0_TIPO_FASES_PRODUCAO	TBU0_TIPO_FASES_PRODUCAO



## 1.3.12 Tabela TIPO\_OPERACOES

## 1.3.12.1 Card of table TIPO\_OPERACOES

Name	TIPO_OPERACOES
Code	TIPO_OPERACOES
Comment	Tabela que armazena os tipos de operações da ficha técnica (Verbos)
Primary Key	PK_TIPO_OPERACOES

## 1.3.12.2 Lista de Colunas da Tabela TIPO\_OPERACOES

Name	Data Type	Domain	Comment
PK_TIPO_O PERACOES	SMALLINT	VALORS	Código do Tipo de Operacao
DSC_OPE	VARCHAR(30)	DESCRICAO_30RQ	Descrição da Operação
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

## 1.3.12.3 Dependências da Tabela TIPO\_OPERACOES

Name	Code	Class Name
FK_OPERACOES_TIPO_OPERAC AO	FK_OPERACOES_TIPO_OPERAC AO	Reference

## 1.3.12.4 Lista de Chaves da Tabela TIPO\_OPERACOES

Name	Code	Primary
PK_TIPO_OPERACOES	PK_TIPO_OPERACOES	X

## 1.3.12.5 Chave PK\_TIPO\_OPERACOES da Tabela TIPO\_OPERACOES

## 1.3.12.5.1 Lista de Colunas da Chave PK\_TIPO\_OPERACOES

Name	Code
PK_TIPO_OPERACOES	PK_TIPO_OPERACOES

## 1.3.12.5.2 Dependências da Chave PK\_TIPO\_OPERACOES

Name	Code	Class Name
PK_TIPO_OPERACOES	PK_TIPO_OPERACOES	Index

## 1.3.12.6 Lista de Índices TIPO\_OPERACOES

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_TIPO_OPE RACOES	PK_TIPO_OPE RACOES	X		X		

**1.3.12.7 Índice PK\_TIPO\_OPERACOES da Tabela TIPO\_OPERACOES****1.3.12.7.1 Lista de Colunas do Índice PK\_TIPO\_OPERACOES**

<b>Name</b>	<b>Code</b>	<b>Sort</b>
PK_TIPO_OPERACOES	PK_TIPO_OPERACOES	Ascending

**1.3.12.8 Lista de Triggers da Tabela TIPO\_OPERACOES**

<b>Name</b>	<b>Code</b>
TBIG_TIPO_OPERACOES	TBIG_TIPO_OPERACOES
TBU0_TIPO_OPERACOES	TBU0_TIPO_OPERACOES

## 1.4 Stored Procedures

### 1.4.1 Procedure STP\_APECAS\_IS\_CHILD\_OF

#### 1.4.1.1 Card of procedure STP\_APECAS\_IS\_CHILD\_OF

<i>Name</i>	STP_APECAS_IS_CHILD_OF
<i>Code</i>	STP_APECAS_IS_CHILD_OF
<i>Parent</i>	Package 'SysTec'
<i>Type</i>	Procedure

#### 1.4.1.2 Code preview of procedure STP\_APECAS\_IS\_CHILD\_OF

set term ^;

```
create procedure STP_APECAS_IS_CHILD_OF (
  P_FK_PRODUTOS_PECAS integer,
  P_FK_PRODUTOS_PECAS_MNT integer
)
returns (
  R_RESULT integer
)
as
  declare variable tmpResult integer;
  declare variable TotalChildren integer;
  declare variable FkPecasMontagem integer;
  declare variable FkFichaTecnicaMontagem integer;
begin
  if ((P_FK_PRODUTOS_PECAS < 1) or (P_FK_PRODUTOS_PECAS_MNT < 1) or
    (P_FK_PRODUTOS_PECAS = P_FK_PRODUTOS_PECAS_MNT)) then
  begin
    R_RESULT = 0;
    suspend;
    exit;
  end
  else
  begin
    TotalChildren = 0;
    select count(*)
      from PECAS_COMPONENTES
     where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS
        and FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS_MNT
      into :TotalChildren;
    if ((TotalChildren is null) or (TotalChildren = 0)) then
    begin
      for select FK_PRODUTOS_PECAS_MNT
        from PECAS_COMPONENTES
       where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS
          and FK_PRODUTOS_PECAS <> FK_PRODUTOS_PECAS_MNT
        into :FkPecasMontagem do
      begin
        if ((tmpResult is not null) and (tmpResult > 0)) then
        begin
          R_RESULT = tmpResult;
          suspend;
          exit;
        end
      end
    end
  end
end
```

```
R_RESULT = 0;
suspend;
exit;
end
else
begin
R_RESULT = :P_FK_PRODUTOS_PECAS;
suspend;
exit;
end
end
end ^

set term ;^

grant execute on procedure STP_APECAS_IS_CHILD_OF to public;
```

#### **1.4.1.3 Text of procedure STP\_APECAS\_IS\_CHILD\_OF**

```
set term ^;

create procedure %PROC% (
P_FK_PRODUTOS_PECAS integer,
P_FK_PRODUTOS_PECAS_MNT integer
)
returns (
R_RESULT integer
)
as
declare variable tmpResult integer;
declare variable TotalChildren integer;
declare variable FkPecasMontagem integer;
declare variable FkFichaTecnicaMontagem integer;
begin
if ((P_FK_PRODUTOS_PECAS < 1) or (P_FK_PRODUTOS_PECAS_MNT < 1) or
(P_FK_PRODUTOS_PECAS = P_FK_PRODUTOS_PECAS_MNT)) then
begin
R_RESULT = 0;
suspend;
exit;
end
else
begin
TotalChildren = 0;
select count(*)
from PECAS_COMPONENTES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS
and FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS_MNT
into :TotalChildren;
if ((TotalChildren is null) or (TotalChildren = 0)) then
begin
for select FK_PRODUTOS_PECAS_MNT
from PECAS_COMPONENTES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS
and FK_PRODUTOS_PECAS <> FK_PRODUTOS_PECAS_MNT
into :FkPecasMontagem do
begin
```

```

if ((tmpResult is not null) and (tmpResult > 0)) then
begin
  R_RESULT = tmpResult;
  suspend;
  exit;
end
end
R_RESULT = 0;
suspend;
exit;
end
else
begin
  R_RESULT = :P_FK_PRODUTOS_PECAS;
  suspend;
  exit;
end
end
end ^

```

set term ^;

grant execute on procedure %PROC% to public;

#### **1.4.1.4 List of diagrams containing the procedure STP\_APECAS IS CHILD OF**

<i>Name</i>	<i>Code</i>
SysTec	SYSTEC

#### **1.4.2 Procedure STP\_CALC\_MATERIALS\_COST**

##### **1.4.2.1 Card of procedure STP\_CALC\_MATERIALS\_COST**

<i>Name</i>	STP_CALC_MATERIALS_COST
<i>Code</i>	STP_CALC_MATERIALS_COST
<i>Parent</i>	Package 'SysTec'
<i>Type</i>	Procedure

##### **1.4.2.2 Code preview of procedure STP\_CALC\_MATERIALS\_COST**

set term ^;

```

create procedure STP_CALC_MATERIALS_COST (
  P_FK_EMPRESAS integer,
  P_FK_TIPO_DETALHE integer,
  P_FLAG_TDET integer,
  P_FLAG_TCUSTO smallint,
  P_FLAG_TACABM smallint,
  P_QTD_DET double precision
)
returns (
  R_COD_REF varchar(20),
  R_DSC_DET varchar(50),
  R_CUST_DET double precision
)
as
declare variable CustMed double precision;
declare variable CustFinal double precision;

```

```
declare variable CustForn    double precision;
declare variable CustReal    double precision;
declare variable CustMedMax  double precision;
declare variable CustFinalMax double precision;
declare variable CustFornMax double precision;
declare variable CustRealMax double precision;
declare variable CustFinalMin double precision;
declare variable CustFornMin double precision;
declare variable CustRealMin double precision;
declare variable CustMedMin  double precision;
declare variable CustoCalc   double precision;
begin
/* Map of P_FlagTCusto
0 ==>
1 ==> CustMed;
2 ==> CustReal;
3 ==> CustFinal;
4 ==> CustForn;
*/
R_CUST_DET = 0;
if ((P_FLAG_TDET < 0) or (P_FLAG_TDET > 2)) then exit;
select Prd.DSC_PROD, Pcd.COD_REF, Prc.CUST_FINAL,
       Prc.CUST_FORN, Prc.CUST_REAL, Prc.CUST_MED
from PRODUTOS Prd, PRODUTOS_CODIGOS Pcd, PRODUTOS_CUSTOS Prc
where Prd.PK_PRODUTOS = :P_FK_TIPO_DETALHE
and Prd.FLAG_ATV      = 0
and Pcd.FK_PRODUTOS = Prd.PK_PRODUTOS
and Pcd.FLAG_TCODE   = 0
and Prc.FK_EMPRESAS = :P_FK_EMPRESAS
and Prc.FK_PRODUTOS = :P_FK_TIPO_DETALHE
into :R_DSC_DET, :R_COD_REF, :CustFinal,
     :CustForn, :CustReal, :CustMed;
if (P_FLAG_TCUSTO = 0) then
suspend;
if (P_FLAG_TCUSTO = 1) then
CustoCalc = CustMed;
if (P_FLAG_TCUSTO = 2) then
CustoCalc = CustReal;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFinal;
if (P_FLAG_TCUSTO = 4) then
CustoCalc = CustForn;
if ((P_FLAG_TDET = 1) and (P_FLAG_TCUSTO > 0)) then
begin
R_Cod_Ref = 'All';
select Min(Pct.CUST_FINAL), Avg(Pct.CUST_FINAL), Max(Pct.CUST_FINAL),
       Min(Pct.CUST_FORN), Avg(Pct.CUST_FORN), Max(Pct.CUST_FORN),
       Min(Pct.CUST_REAL), Avg(Pct.CUST_REAL), Max(Pct.CUST_REAL),
       Min(Pct.CUST_MED), Avg(Pct.CUST_MED), Max(Pct.CUST_MED)
from PRODUTOS Prd, PRODUTOS_COMPRAS Pcm, PRODUTOS_CUSTOS Pct
where Prd.PK_PRODUTOS      > 0
and Prd.FLAG_ATV          = 0
and Pcm.FK_PRODUTOS       = Prd.PK_PRODUTOS
and Pcm.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_DETALHE
and Pct.FK_EMPRESAS       = :P_FK_EMPRESAS
and Pct.FK_PRODUTOS       = Prd.PK_PRODUTOS
into :CustFinalMin, :CustFinal, :CustFinalMax,
     :CustFornMin, :CustForn, :CustFornMax,
     :CustRealMin, :CustReal, :CustRealMax,
```

```

:CustMedMin , :CustMed , :CustMedMax;
select DSC_ACABM
from TIPO_ACABAMENTOS
where PK_TIPO_ACABAMENTOS = :P_FK_TIPO_DETALHE
into :R_DSC_DET;
/*
0 ==> Menor Valor dos acabamentos do mesmo tipo
1 ==> Valor Médio dos acabamentos do mesmo tipo
2 ==> Maior Valor dos acabamentos do mesmo tipo
*/
if (P_FLAG_TACABM = 0) then
begin
if (P_FLAG_TCUSTO = 1) then
CustoCalc = CustMedMin;
if (P_FLAG_TCUSTO = 2) then
CustoCalc = CustRealMin;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFinalMin;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFornMin;
end
if (P_FLAG_TACABM = 1) then
begin
if (P_FLAG_TCUSTO = 1) then
CustoCalc = CustMed;
if (P_FLAG_TCUSTO = 2) then
CustoCalc = CustReal;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFinal;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustForn;
end
end
if (P_FLAG_TACABM = 0) then
begin
if (P_FLAG_TCUSTO = 1) then
CustoCalc = CustMedMax;
if (P_FLAG_TCUSTO = 2) then
CustoCalc = CustRealMax;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFinalMax;
if (P_FLAG_TCUSTO = 3) then
CustoCalc = CustFornMax;
end
end
end
R_CUST_DET = CustoCalc * P_QTD_DET;
suspend;
end^

set term ^;

grant execute on procedure STP_CALC_MATERIALS_COST to public;

```

#### **1.4.2.3 Text of procedure STP\_CALC\_MATERIALS\_COST**

```

set term ^;

create procedure %PROC% (
P_FK_EMPRESAS integer,
P_FK_TIPO_DETALHE integer,
P_FLAG_TDET integer,

```

```
P_FLAG_TCUSTO    smallint,  
P_FLAG_TACABM    smallint,  
P_QTD_DET        double precision  
)  
returns (  
    R_COD_REF  varchar(20),  
    R_DSC_DET  varchar(50),  
    R_CUST_DET double precision  
)  
as  
declare variable CustMed    double precision;  
declare variable CustFinal  double precision;  
declare variable CustForn   double precision;  
declare variable CustReal   double precision;  
declare variable CustMedMax  double precision;  
declare variable CustFinalMax double precision;  
declare variable CustFornMax double precision;  
declare variable CustRealMax double precision;  
declare variable CustFinalMin double precision;  
declare variable CustFornMin double precision;  
declare variable CustRealMin double precision;  
declare variable CustMedMin  double precision;  
declare variable CustoCalc   double precision;  
begin  
/* Map of P_FlagTCusto  
0 ==>  
1 ==> CustMed;  
2 ==> CustReal;  
3 ==> CustFinal;  
4 ==> CustForn;  
*/  
R_CUST_DET = 0;  
if ((P_FLAG_TDET < 0) or (P_FLAG_TDET > 2)) then exit;  
select Prd.DSC_PROD, Pcd.COD_REF, Prc.CUST_FINAL,  
       Prc.CUST_FORN, Prc.CUST_REAL, Prc.CUST_MED  
from PRODUTOS Prd, PRODUTOS_CODIGOS Pcd, PRODUTOS_CUSTOS Prc  
where Prd.PK_PRODUTOS = :P_FK_TIPO_DETALHE  
and Prd.FLAG_ATV = 0  
and Pcd.FK_PRODUTOS = Prd.PK_PRODUTOS  
and Pcd.FLAG_TCODE = 0  
and Prc.FK_EMPRESAS = :P_FK_EMPRESAS  
and Prc.FK_PRODUTOS = :P_FK_TIPO_DETALHE  
into :R_DSC_DET, :R_COD_REF, :CustFinal,  
     :CustForn, :CustReal, :CustMed;  
if (P_FLAG_TCUSTO = 0) then  
    suspend;  
if (P_FLAG_TCUSTO = 1) then  
    CustoCalc = CustMed;  
if (P_FLAG_TCUSTO = 2) then  
    CustoCalc = CustReal;  
if (P_FLAG_TCUSTO = 3) then  
    CustoCalc = CustFinal;  
if (P_FLAG_TCUSTO = 4) then  
    CustoCalc = CustForn;  
if ((P_FLAG_TDET = 1) and (P_FLAG_TCUSTO > 0)) then  
begin  
    R_Cod_Ref = 'All';  
select Min(Pct.CUST_FINAL), Avg(Pct.CUST_FINAL), Max(Pct.CUST_FINAL),  
       Min(Pct.CUST_FORN) , Avg(Pct.CUST_FORN) , Max(Pct.CUST_FORN),
```



```

    Min(Pct.CUST_REAL) , Avg(Pct.CUST_REAL) , Max(Pct.CUST_REAL),
    Min(Pct.CUST_MED) , Avg(Pct.CUST_MED) , Max(Pct.CUST_MED)
from PRODUTOS Prd, PRODUTOS_COMPRAS Pcm, PRODUTOS_CUSTOS Pct
where Prd.PK_PRODUTOS      > 0
and Prd.FLAG_ATV           = 0
and Pcm.FK_PRODUTOS        = Prd.PK_PRODUTOS
and Pcm.FK_TIPO_ACABAMENTOS = :P_FK_TIPO_DETALHE
and Pct.FK_EMPRESAS        = :P_FK_EMPRESAS
and Pct.FK_PRODUTOS        = Prd.PK_PRODUTOS
into :CustFinalMin, :CustFinal, :CustFinalMax,
    :CustFornMin , :CustForn , :CustFornMax,
    :CustRealMin , :CustReal , :CustRealMax,
    :CustMedMin , :CustMed , :CustMedMax;
select DSC_ACABM
from TIPO_ACABAMENTOS
where PK_TIPO_ACABAMENTOS = :P_FK_TIPO_DETALHE
into :R_DSC_DET;
/*
0 ==> Menor Valor dos acabamentos do mesmo tipo
1 ==> Valor Médio dos acabamentos do mesmo tipo
2 ==> Maior Valor dos acabamentos do mesmo tipo
*/
if (P_FLAG_TACABM = 0) then
begin
    if (P_FLAG_TCUSTO = 1) then
        CustoCalc = CustMedMin;
    if (P_FLAG_TCUSTO = 2) then
        CustoCalc = CustRealMin;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustFinalMin;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustFornMin;
    end
if (P_FLAG_TACABM = 1) then
begin
    if (P_FLAG_TCUSTO = 1) then
        CustoCalc = CustMed;
    if (P_FLAG_TCUSTO = 2) then
        CustoCalc = CustReal;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustFinal;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustForn;
    end
if (P_FLAG_TACABM = 0) then
begin
    if (P_FLAG_TCUSTO = 1) then
        CustoCalc = CustMedMax;
    if (P_FLAG_TCUSTO = 2) then
        CustoCalc = CustRealMax;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustFinalMax;
    if (P_FLAG_TCUSTO = 3) then
        CustoCalc = CustFornMax;
    end
end
R_CUST_DET = CustoCalc * P_QTD_DET;
suspend;
end^

```

set term ;^

#### 1.4.2.4 List of diagrams containing the procedure STP\_CALC\_MATERIALS\_COST

Name	Code
SysTec	SYSTEC

#### 1.4.3 Procedure STP\_COPY\_FICHA\_TECNICA

##### 1.4.3.1 Card of procedure STP\_COPY\_FICHA\_TECNICA

Name	STP_COPY_FICHA_TECNICA
Code	STP_COPY_FICHA_TECNICA
Parent	Package 'SysTec'
Type	Procedure

##### 1.4.3.2 Code preview of procedure STP\_COPY\_FICHA\_TECNICA

set term ^;

```
create procedure STP_COPY_FICHA_TECNICA (
  P_FK_PRODUTOS_PECAS_FROM integer,
  P_FK_PRODUTOS_PECAS_TO integer,
  P_MAJ_VER_TO smallint,
  P_MIN_VER_TO smallint
)
```

returns

```
(
  R_TOTAL_COMPONENTES_INCLUIDOS integer,
  R_TOTAL_FASES_INCLUIDAS integer,
  R_TOTAL_OPERACOES integer,
  R_TOTAL_PECAS integer,
  R_TOTAL_OPERACOES_DETALHES integer,
  R_TOTAL_MAQUINAS integer,
  R_TOTAL_FERRAMENTAS integer
)
```

as

```
declare variable FkPecasMontagem integer;
declare variable QtdPec float;
declare variable FkTipoFasesProducao integer;
declare variable SeqFase integer;
declare variable FkFasesProducao integer;
declare variable PkFasesProducao integer;
declare variable fkOperacoes integer;
declare variable PkOperacoes integer;
declare variable FkTipoOperacoes integer;
declare variable SeqOpe integer;
declare variable TempoMedio float;
declare variable SeqMont integer;
declare variable FkTipoDetalhe integer;
declare variable FlagTDet integer;
declare variable QtdDet float;
declare variable FkMaquinas integer;
declare variable TmpStp float;
declare variable TmpOper float;
declare variable FlagDef integer;
declare variable FkInsumos integer;
```

```
declare variable FkTipoDocumentos    integer;
declare variable CodRef                varchar(20);
declare variable AltMax                float;
declare variable LargMax              float;
declare variable ProfMax              float;
declare variable AltMin               float;
declare variable LargMin              float;
declare variable ProfMin              float;
declare variable PercPerda            float;
declare variable ThisPecaIsAChild     integer;
declare variable FlagAtivDestino      integer;
declare variable FlagOpDestino        integer;
declare variable FkFichaTecnicaTemplate integer;
declare variable FkFichaTecnicaDestino integer;
declare variable HasRecords           smallint;
begin
  R_TOTAL_COMPONENTES_INCLUIDOS = 0;
  R_TOTAL_FASES_INCLUIDAS      = 0;
  R_TOTAL_OPERACOES           = 0;
  R_TOTAL_PECAS               = 0;
  R_TOTAL_MQUINAS             = 0;
  R_TOTAL_FERRAMENTAS         = 0;
  select count(*) from PRODUTOS_PECAS
  where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
  into :R_TOTAL_PECAS;
  if (R_TOTAL_PECAS < 1) then
    begin
      R_TOTAL_COMPONENTES_INCLUIDOS = -7; /* Peca Destino nao existe */
      suspend;
      exit;
    end
  R_TOTAL_PECAS = 1;
  select count(*) from OPERACOES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
  into :R_TOTAL_OPERACOES;
  if (R_TOTAL_OPERACOES < 1) then
    begin
      R_TOTAL_COMPONENTES_INCLUIDOS = -6; /* Ficha Tecnica Origem Vazia ou peça de origem não existe */
      suspend;
      exit;
    end
  ThisPecaIsAChild = 0;
  select R_RESULT
  from STP_APECAS_IS_CHILD_OF(
    :P_FK_PRODUTOS_PECAS_FROM, :P_FK_PRODUTOS_PECAS_TO)
  into :ThisPecaIsAChild;
  if ((ThisPecaIsAChild is not null) and (ThisPecaIsAChild > 0)) then
    begin
      R_TOTAL_COMPONENTES_INCLUIDOS = -8; /* Peca Destino ja e filha da peca origem */
      suspend;
      exit;
    end
  FkFichaTecnicaTemplate = -1;
  select FK_PRODUTOS
  from PRODUTOS_PECAS
  where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
  into :FkFichaTecnicaTemplate;
  if ((FkFichaTecnicaTemplate is null) or (FkFichaTecnicaTemplate < 1)) then
    begin
```

```
FkFichaTecnicaDestino = -1;
select Gen_Id(PRODUTOS, 1) from PARAMETRO_GLOBAIS
into :P_FK_PRODUTOS_PECAS_TO;
insert into PRODUTOS
(PK_PRODUTOS, FK_SECOES, FK_GRUPOS, FK_SUBGRUPOS, FK_UNIDADES,
DSC_PROD, FLAG_ATV, FAT_CONV, QTD_UNI)
select :P_FK_PRODUTOS_PECAS_TO, FK_SECOES, FK_GRUPOS, FK_SUBGRUPOS,
FK_UNIDADES, DSC_PROD, 1, 0.0, QTD_UNI
from PRODUTOS
where PK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
insert into PRODUTOS_CODIGOS
(FK_PRODUTOS, PK_PRODUTOS_CODIGOS, COD_REF, DSC_CODE,
FLAG_TCODE, FLAG_TBARCODE)
select :P_FK_PRODUTOS_PECAS_TO, PK_PRODUTOS_CODIGOS, COD_REF,
'Referência', FLAG_TCODE, FLAG_TBARCODE
from PRODUTOS_CODIGOS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
select Count(*) from PRODUTOS_IMAGENS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM
into HasRecords;
if ((HasRecords is not null) or (HasRecords > 0)) then
begin
insert into PRODUTOS_IMAGENS
(FK_PRODUTOS, IMG_PROD, FLAG_TIMG)
select :P_FK_PRODUTOS_PECAS_TO, IMG_PROD, FLAG_TIMG
from PRODUTOS_IMAGENS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
end
HasRecords = 0;
select Count(*) from PRODUTOS_CARACTERISTICAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM
into HasRecords;
if ((HasRecords is not null) or (HasRecords > 0)) then
begin
insert into PRODUTOS_CARACTERISTICAS
(FK_PRODUTOS, CRT_PROD)
select :P_FK_PRODUTOS_PECAS_TO, CRT_PROD
from PRODUTOS_CARACTERISTICAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
end
insert into PRODUTOS_PECAS
(FK_PRODUTOS, MAJ_VER, MIN_VER, MOT_NVER, ALT_PEC,
PROF_PEC, LARG_PEC, FLAG_TCOMP, FLAG_OP)
select :P_FK_PRODUTOS_PECAS_TO, :P_MAJ_VER_TO, :P_MIN_VER_TO,
'COPIA DE FICHA TECNICA', ALT_PEC,
PROF_PEC, LARG_PEC, FLAG_TCOMP, FLAG_OP
from PRODUTOS_PECAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
end
else
begin
FlagAtivDestino = -1;
FlagOpDestino = -1;
select Prd.FLAG_ATV, Ppc.FLAG_OP
from PRODUTOS Prd, PRODUTOS_PECAS Ppc
where Prd.PK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
and Ppc.FK_PRODUTOS = Prd.PK_PRODUTOS
into :FlagAtivDestino, :FlagOpDestino;
if ((FlagAtivDestino is not null) and (FlagAtivDestino = 1)) then
```

```
begin
  R_TOTAL_COMPONENTES_INCLUIDOS = -11; /* Ficha Tecnica de Destino esta ativa */
  suspend;
  exit;
end
if ((FlagOpDestino is not null) and (FlagOpDestino = 1)) then
begin
  R_TOTAL_COMPONENTES_INCLUIDOS = -12; /* Ficha Tecnica de Destino tem OP */
  suspend;
  exit;
end
delete from PECAS_FERRAMENTAS
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_MAQUINAS
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from OPERACOES_DETALHES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_COMPO_OPER
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from OPERACOES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from FASES_PRODUCAO
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_COMPONENTES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
update PRODUTOS_PECAS set
  MOT_NVER = 'COPIA DE FICHA TECNICA'
  where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO;
end
select QTD_PEC from PECAS_COMPONENTES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
    and FK_PRODUTOS_PECAS_MNT = FK_PRODUTOS_PECAS_MNT
  into :QtdPec;
insert into PECAS_COMPONENTES
  (FK_PRODUTOS_PECAS, FK_PRODUTOS_PECAS_MNT, QTD_PEC)
values
  (:P_FK_PRODUTOS_PECAS_TO, :P_FK_PRODUTOS_PECAS_TO, :QtdPec);
R_TOTAL_COMPONENTES_INCLUIDOS = R_TOTAL_COMPONENTES_INCLUIDOS + 1;
for select FK_PRODUTOS_PECAS_MNT, QTD_PEC
  from PECAS_COMPONENTES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
    and FK_PRODUTOS_PECAS <> FK_PRODUTOS_PECAS_MNT
  into :FkPecasMontagem, :QtdPec do
begin
  insert into PECAS_COMPONENTES
    (FK_PRODUTOS_PECAS, FK_PRODUTOS_PECAS_MNT, QTD_PEC)
  values
    (:P_FK_PRODUTOS_PECAS_TO, :FkPecasMontagem, :QtdPec);
  R_TOTAL_COMPONENTES_INCLUIDOS = R_TOTAL_COMPONENTES_INCLUIDOS + 1;
end
for select PK_FASES_PRODUCAO, FK_TIPO_FASES_PRODUCAO, SEQ_FASE
  from FASES_PRODUCAO
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
  into :FkFasesProducao, :FkTipoFasesProducao, :SeqFase do
begin
  insert into FASES_PRODUCAO
    (FK_PRODUTOS_PECAS, FK_TIPO_FASES_PRODUCAO, SEQ_FASE)
  values
    (:P_FK_PRODUTOS_PECAS_TO, :FkTipoFasesProducao, :SeqFase);
```

```
select max(PK_FASES_PRODUCAO)
from FASES_PRODUCAO
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO
into :PkFasesProducao;
R_TOTAL_FASES_INCLUIDAS = R_TOTAL_FASES_INCLUIDAS + 1;
for select PK_OPERACOES, FK_TIPO_OPERACOES, FK_TIPO_DOCUMENTOS, SEQ_OPE,
    COD_REF, ALT_MAX, LARG_MAX, PROF_MAX, ALT_MIN,
    LARG_MIN, PROF_MIN, TEMPO_MEDIO
from OPERACOES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
into :fkOperacoes, :FkTipoOperacoes, :FkTipoDocumentos, :SeqOpe,
    :CodRef, :AltMax, :LargMax, :ProfMax, :AltMin,
    :LargMin, :ProfMin, :TempoMedio do
begin
insert into OPERACOES
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_TIPO_OPERACOES,
FK_TIPO_DOCUMENTOS, SEQ_OPE, COD_REF, ALT_MAX, LARG_MAX,
PROF_MAX, ALT_MIN, LARG_MIN, PROF_MIN, TEMPO_MEDIO)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :FkTipoOperacoes,
:FkTipoDocumentos, :SeqOpe, :CodRef, :AltMax, :LargMax,
:ProfMax, :AltMin, :LargMin, :ProfMin, :TempoMedio);
select max(PK_OPERACOES)
from OPERACOES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO
and FK_FASES_PRODUCAO = :PkFasesProducao
into :PkOperacoes;
R_TOTAL_OPERACOES = R_TOTAL_OPERACOES + 1;
for select FK_PRODUTOS_PECAS_MNT, SEQ_MONT, PERC_PERDA
from PECAS_COMPO_OPER
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :FkOperacoes
into :FkPecasMontagem, :SeqMont, :PercPerda do
begin
insert into PECAS_COMPO_OPER
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
FK_PRODUTOS_PECAS_MNT, SEQ_MONT, PERC_PERDA)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
:FkPecasMontagem, :SeqMont, :PercPerda);
R_TOTAL_PECAS = R_TOTAL_PECAS + 1;
end
for select FK_TIPO_DETALHE, FLAG_TDET, QTD_DET, PERC_PERDA
from OPERACOES_DETALHES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :fkOperacoes
into :FkTipoDetalhe, :FlagTDet, :QtdDet, :PercPerda do
begin
insert into OPERACOES_DETALHES
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_TIPO_DETALHE,
FK_OPERACOES, FLAG_TDET, QTD_DET, PERC_PERDA)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :FkTipoDetalhe,
:PkOperacoes, :FlagTDet, :QtdDet, :PercPerda);
R_TOTAL_OPERACOES_DETALHES = R_TOTAL_OPERACOES_DETALHES + 1;
end
```

```

for select FK_PRODUTOS_MAQUINAS, TMP_STP, TMP_OPER, FLAG_DEF
  from PECAS_MAQUINAS
 where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
 and FK_FASES_PRODUCAO = :FkFasesProducao
 and FK_OPERACOES      = :FkOperacoes
 into :FkMaquinas, :TmpStp, :TmpOper, :FlagDef do
begin
  insert into PECAS_MAQUINAS
    (FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
     FK_PRODUTOS_MAQUINAS, TMP_STP, TMP_OPER, FLAG_DEF)
  values
    (:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
     :FkMaquinas, :TmpStp, :TmpOper, :FlagDef);
  R_TOTAL_MAQUINAS = R_TOTAL_MAQUINAS + 1;
end
for select FK_PRODUTOS_MAQUINAS, FK_PRODUTOS_COMPRAS
  from PECAS_FERRAMENTAS
 where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
 and FK_FASES_PRODUCAO = :FkFasesProducao
 and FK_OPERACOES      = :fkOperacoes
 into :FkMaquinas, :FkInsumos do
begin
  insert into PECAS_FERRAMENTAS
    (FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
     FK_PRODUTOS_MAQUINAS, FK_PRODUTOS_COMPRAS)
  values
    (:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
     :FkMaquinas, :FkInsumos);
  R_TOTAL_FERRAMENTAS = R_TOTAL_FERRAMENTAS + 1;
end
end
end
suspend;
end ^

```

set term ^;

grant execute on procedure STP\_COPY\_FICHA\_TECNICA to public;

#### **1.4.3.3 Text of procedure STP\_COPY\_FICHA\_TECNICA**

set term ^;

```

create procedure %PROC% (
  P_FK_PRODUTOS_PECAS_FROM integer,
  P_FK_PRODUTOS_PECAS_TO   integer,
  P_MAJ_VER_TO             smallint,
  P_MIN_VER_TO             smallint
)
returns
(
  R_TOTAL_COMPONENTES_INCLUIDOS integer,
  R_TOTAL_FASES_INCLUIDAS      integer,
  R_TOTAL_OPERACOES            integer,
  R_TOTAL_PECAS                integer,
  R_TOTAL_OPERACOES_DETALHES   integer,
  R_TOTAL_MAQUINAS             integer,
  R_TOTAL_FERRAMENTAS          integer
)
as

```

```
declare variable FkPecasMontagem      integer;
declare variable QtdPec                float;
declare variable FkTipoFasesProducao  integer;
declare variable SeqFase                integer;
declare variable FkFasesProducao      integer;
declare variable PkFasesProducao      integer;
declare variable fkOperacoes          integer;
declare variable PkOperacoes          integer;
declare variable FkTipoOperacoes     integer;
declare variable SeqOpe                integer;
declare variable TempoMedio           float;
declare variable SeqMont               integer;
declare variable FkTipoDetalhe        integer;
declare variable FlagTDet              integer;
declare variable QtdDet                float;
declare variable FkMaquinas            integer;
declare variable TmpStp                float;
declare variable TmpOper               float;
declare variable FlagDef                integer;
declare variable FkInsumos             integer;
declare variable FkTipoDocumentos     integer;
declare variable CodRef                varchar(20);
declare variable AltMax                float;
declare variable LargMax               float;
declare variable ProfMax               float;
declare variable AltMin                float;
declare variable LargMin              float;
declare variable ProfMin               float;
declare variable PercPerda             float;
declare variable ThisPecaIsAChild      integer;
declare variable FlagAtivDestino       integer;
declare variable FlagOpDestino         integer;
declare variable FkFichaTecnicaTemplate integer;
declare variable FkFichaTecnicaDestino integer;
declare variable HasRecords            smallint;
begin
  R_TOTAL_COMPONENTES_INCLUIDOS = 0;
  R_TOTAL_FASES_INCLUIDAS      = 0;
  R_TOTAL_OPERACOES            = 0;
  R_TOTAL_PECAS                = 0;
  R_TOTAL_MAUQUINAS            = 0;
  R_TOTAL_FERRAMENTAS          = 0;
  select count(*) from PRODUTOS_PECAS
  where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
  into :R_TOTAL_PECAS;
  if (R_TOTAL_PECAS < 1) then
    begin
      R_TOTAL_COMPONENTES_INCLUIDOS = -7; /* Peca Destino nao existe */
      suspend;
      exit;
    end
  R_TOTAL_PECAS = 1;
  select count(*) from OPERACOES
  where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
  into :R_TOTAL_OPERACOES;
  if (R_TOTAL_OPERACOES < 1) then
    begin
      R_TOTAL_COMPONENTES_INCLUIDOS = -6; /* Ficha Tecnica Origem Vazia ou peça de origem não existe */
      suspend;
```



```
exit;
end
ThisPecaIsAChild = 0;
select R_RESULT
from STP_APECAS_IS_CHILD_OF(
:P_FK_PRODUTOS_PECAS_FROM, :P_FK_PRODUTOS_PECAS_TO)
into :ThisPecaIsAChild;
if ((ThisPecaIsAChild is not null) and (ThisPecaIsAChild > 0)) then
begin
R_TOTAL_COMPONENTES_INCLUIDOS = -8; /* Peca Destino ja e filha da peca origem */
suspend;
exit;
end
FkFichaTecnicaTemplate = -1;
select FK_PRODUTOS
from PRODUTOS_PECAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
into :FkFichaTecnicaTemplate;
if ((FkFichaTecnicaTemplate is null) or (FkFichaTecnicaTemplate < 1)) then
begin
FkFichaTecnicaDestino = -1;
select Gen_Id(PRODUTOS, 1) from PARAMETRO_GLOBAIS
into :P_FK_PRODUTOS_PECAS_TO;
insert into PRODUTOS
(PK_PRODUTOS, FK_SECOES, FK_GRUPOS, FK_SUBGRUPOS, FK_UNIDADES,
DSC_PROD, FLAG_ATV, FAT_CONV, QTD_UNI)
select :P_FK_PRODUTOS_PECAS_TO, FK_SECOES, FK_GRUPOS, FK_SUBGRUPOS,
FK_UNIDADES, DSC_PROD, 1, 0.0, QTD_UNI
from PRODUTOS
where PK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
insert into PRODUTOS_CODIGOS
(FK_PRODUTOS, PK_PRODUTOS_CODIGOS, COD_REF, DSC_CODE,
FLAG_TCODE, FLAG_TBARCODE)
select :P_FK_PRODUTOS_PECAS_TO, PK_PRODUTOS_CODIGOS, COD_REF,
'Referência', FLAG_TCODE, FLAG_TBARCODE
from PRODUTOS_CODIGOS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
select Count(*) from PRODUTOS_IMAGENS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM
into HasRecords;
if ((HasRecords is not null) or (HasRecords > 0)) then
begin
insert into PRODUTOS_IMAGENS
(FK_PRODUTOS, IMG_PROD, FLAG_TIMG)
select :P_FK_PRODUTOS_PECAS_TO, IMG_PROD, FLAG_TIMG
from PRODUTOS_IMAGENS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
end
HasRecords = 0;
select Count(*) from PRODUTOS_CARACTERISTICAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM
into HasRecords;
if ((HasRecords is not null) or (HasRecords > 0)) then
begin
insert into PRODUTOS_CARACTERISTICAS
(FK_PRODUTOS, CRT_PROD)
select :P_FK_PRODUTOS_PECAS_TO, CRT_PROD
from PRODUTOS_CARACTERISTICAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
```

```
end
insert into PRODUTOS_PECAS
(FK_PRODUTOS, MAJ_VER, MIN_VER, MOT_NVER, ALT_PEC,
 PROF_PEC, LARG_PEC, FLAG_TCOMP, FLAG_OP)
select :P_FK_PRODUTOS_PECAS_TO, :P_MAJ_VER_TO, :P_MIN_VER_TO,
       'COPIA DE FICHA TECNICA', ALT_PEC,
       PROF_PEC, LARG_PEC, FLAG_TCOMP, FLAG_OP
from PRODUTOS_PECAS
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_FROM;
end
else
begin
  FlagAtivDestino = -1;
  FlagOpDestino = -1;
  select Prd.FLAG_ATV, Ppc.FLAG_OP
  from PRODUTOS Prd, PRODUTOS_PECAS Ppc
  where Prd.PK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO
  and Ppc.FK_PRODUTOS = Prd.PK_PRODUTOS
  into :FlagAtivDestino, :FlagOpDestino;
  if ((FlagAtivDestino is not null) and (FlagAtivDestino = 1)) then
  begin
    R_TOTAL_COMPONENTES_INCLUIDOS = -11; /* Ficha Tecnica de Destino esta ativa */
    suspend;
  exit;
  end
  if ((FlagOpDestino is not null) and (FlagOpDestino = 1)) then
  begin
    R_TOTAL_COMPONENTES_INCLUIDOS = -12; /* Ficha Tecnica de Destino tem OP */
    suspend;
  exit;
  end
delete from PECAS_FERRAMENTAS
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_MAQUINAS
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from OPERACOES_DETALHES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_COMPO_OPER
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from OPERACOES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from FASES_PRODUCAO
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
delete from PECAS_COMPONENTES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO;
update PRODUTOS_PECAS set
  MOT_NVER = 'COPIA DE FICHA TECNICA'
where FK_PRODUTOS = :P_FK_PRODUTOS_PECAS_TO;
end
select QTD_PEC from PECAS_COMPONENTES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_PRODUTOS_PECAS_MNT = FK_PRODUTOS_PECAS_MNT
into :QtdPec;
insert into PECAS_COMPONENTES
(FK_PRODUTOS_PECAS, FK_PRODUTOS_PECAS_MNT, QTD_PEC)
values
(:P_FK_PRODUTOS_PECAS_TO, :P_FK_PRODUTOS_PECAS_TO, :QtdPec);
R_TOTAL_COMPONENTES_INCLUIDOS = R_TOTAL_COMPONENTES_INCLUIDOS + 1;
for select FK_PRODUTOS_PECAS_MNT, QTD_PEC
```

```
from PECAS_COMPONENTES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_PRODUTOS_PECAS <> FK_PRODUTOS_PECAS_MNT
into :FkPecasMontagem, :QtdPec do
begin
insert into PECAS_COMPONENTES
(FK_PRODUTOS_PECAS, FK_PRODUTOS_PECAS_MNT, QTD_PEC)
values
(:P_FK_PRODUTOS_PECAS_TO, :FkPecasMontagem, :QtdPec);
R_TOTAL_COMPONENTES_INCLUIDOS = R_TOTAL_COMPONENTES_INCLUIDOS + 1;
end
for select PK_FASES_PRODUCAO, FK_TIPO_FASES_PRODUCAO, SEQ_FASE
from FASES_PRODUCAO
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
into :FkFasesProducao, :FkTipoFasesProducao, :SeqFase do
begin
insert into FASES_PRODUCAO
(FK_PRODUTOS_PECAS, FK_TIPO_FASES_PRODUCAO, SEQ_FASE)
values
(:P_FK_PRODUTOS_PECAS_TO, :FkTipoFasesProducao, :SeqFase);
select max(PK_FASES_PRODUCAO)
from FASES_PRODUCAO
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO
into :PkFasesProducao;
R_TOTAL_FASES_INCLUIDAS = R_TOTAL_FASES_INCLUIDAS + 1;
for select PK_OPERACOES, FK_TIPO_OPERACOES, FK_TIPO_DOCUMENTOS, SEQ_OPE,
COD_REF, ALT_MAX, LARG_MAX, PROF_MAX, ALT_MIN,
LARG_MIN, PROF_MIN, TEMPO_MEDIO
from OPERACOES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
into :fkOperacoes, :FkTipoOperacoes, :FkTipoDocumentos, :SeqOpe,
:CodRef, :AltMax, :LargMax, :ProfMax, :AltMin,
:LargMin, :ProfMin, :TempoMedio do
begin
insert into OPERACOES
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_TIPO_OPERACOES,
FK_TIPO_DOCUMENTOS, SEQ_OPE, COD_REF, ALT_MAX, LARG_MAX,
PROF_MAX, ALT_MIN, LARG_MIN, PROF_MIN, TEMPO_MEDIO)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :FkTipoOperacoes,
:FkTipoDocumentos, :SeqOpe, :CodRef, :AltMax, :LargMax,
:ProfMax, :AltMin, :LargMin, :ProfMin, :TempoMedio);
select max(PK_OPERACOES)
from OPERACOES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_TO
and FK_FASES_PRODUCAO = :PkFasesProducao
into :PkOperacoes;
R_TOTAL_OPERACOES = R_TOTAL_OPERACOES + 1;
for select FK_PRODUTOS_PECAS_MNT, SEQ_MONT, PERC_PERDA
from PECAS_COMPO_OPER
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :FkOperacoes
into :FkPecasMontagem, :SeqMont, :PercPerda do
begin
insert into PECAS_COMPO_OPER
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
FK_PRODUTOS_PECAS_MNT, SEQ_MONT, PERC_PERDA)
```

```

values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
:FkPecasMontagem, :SeqMont, :PercPerda);
R_TOTAL_PECAS = R_TOTAL_PECAS + 1;
end
for select FK_TIPO_DETALHE, FLAG_TDET, QTD_DET, PERC_PERDA
from OPERACOES_DETALHES
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :fkOperacoes
into :FkTipoDetalhe, :FlagTDet, :QtdDet, :PercPerda do
begin
insert into OPERACOES_DETALHES
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_TIPO_DETALHE,
FK_OPERACOES, FLAG_TDET, QTD_DET, PERC_PERDA)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :FkTipoDetalhe,
:PkOperacoes, :FlagTDet, :QtdDet, :PercPerda);
R_TOTAL_OPERACOES_DETALHES = R_TOTAL_OPERACOES_DETALHES + 1;
end
for select FK_PRODUTOS_MAQUINAS, TMP_STP, TMP_OPER, FLAG_DEF
from PECAS_MAQUINAS
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :FkOperacoes
into :FkMaquinas, :TmpStp, :TmpOper, :FlagDef do
begin
insert into PECAS_MAQUINAS
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
FK_PRODUTOS_MAQUINAS, TMP_STP, TMP_OPER, FLAG_DEF)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
:FkMaquinas, :TmpStp, :TmpOper, :FlagDef);
R_TOTAL_MAQUINAS = R_TOTAL_MAQUINAS + 1;
end
for select FK_PRODUTOS_MAQUINAS, FK_PRODUTOS_COMPRAS
from PECAS_FERRAMENTAS
where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS_PECAS_FROM
and FK_FASES_PRODUCAO = :FkFasesProducao
and FK_OPERACOES = :fkOperacoes
into :FkMaquinas, :FkInsumos do
begin
insert into PECAS_FERRAMENTAS
(FK_PRODUTOS_PECAS, FK_FASES_PRODUCAO, FK_OPERACOES,
FK_PRODUTOS_MAQUINAS, FK_PRODUTOS_COMPRAS)
values
(:P_FK_PRODUTOS_PECAS_TO, :PkFasesProducao, :PkOperacoes,
:FkMaquinas, :FkInsumos);
R_TOTAL_FERRAMENTAS = R_TOTAL_FERRAMENTAS + 1;
end
end
end
suspend;
end ^

set term ;^

```

<i>Name</i>	<i>Code</i>
SysTec	SYSTEC

#### 1.4.4 Procedure STP\_GET\_DATA\_COMPONENT

##### 1.4.4.1 Card of procedure STP\_GET\_DATA\_COMPONENT

<i>Name</i>	STP_GET_DATA_COMPONENT
<i>Code</i>	STP_GET_DATA_COMPONENT
<i>Parent</i>	Package 'SysTec'
<i>Type</i>	Procedure

##### 1.4.4.2 Code preview of procedure STP\_GET\_DATA\_COMPONENT

set term ^;

```

create procedure STP_GET_DATA_COMPONENT (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer
)
returns (
  R_FK_PRODUTOS integer,
  R_DSC_PROD   varchar(50),
  R_DSC_DET    varchar(50),
  R_COD_REF    varchar(20),
  R_COD_REF_DET varchar(20),
  R_SOMA_NIVEL double precision
)
as
  declare variable FkTipoDetalhe integer;
  declare variable PkOperacoes integer;
  declare variable FlagTDet smallint;
  declare variable FlagTCusto smallint;
  declare variable FlagAcbMed smallint;
  declare variable TempoMedio double precision;
  declare variable CustDet double precision;
  declare variable QtdDet double precision;
  declare variable PercPerda double precision;
begin
  select FLAG_TCUSTO, FLAG_TACABM
  from PARAMETROS_ESTQ
  where FK_EMPRESAS = :P_FK_EMPRESAS
  into :FlagTCusto, :FlagAcbMed;
  if (:P_FK_PRODUTOS is not null) then
  begin
    select Pcd.PK_PRODUTOS_CODIGOS, DSC_PROD
    from PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
    where Prd.PK_PRODUTOS = :P_FK_PRODUTOS
    and Pcd.FK_PRODUTOS = Prd.PK_PRODUTOS
    and Pcd.FLAG_TCODE = 0
    and Pcd.FLAG_TBARCODE = 0
    into :R_COD_REF, :R_DSC_PROD;

    for select TEMPO_MEDIO, PK_OPERACOES
    from OPERACOES
    where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
    into :TempoMedio, :PkOperacoes do

```

```
begin
  if (PkOperacoes is not null) then
    begin
      for select FK_TIPO_DETALHE, FLAG_TDET, QTD_DET,
        PERC_PERDA
        from OPERACOES_DETALHES
        where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
          and FK_OPERACOES = :PkOperacoes
          into :FkTipoDetalhe, :FlagTDet, :QtdDet,
            :PercPerda do
        begin
          if (FkTipoDetalhe is not null) then
            select R_COD_REF, R_DSC_DET, R_CUST_DET
              from STP_CALC_MATERIALS_COST (:FkTipoDetalhe, :FlagTDet,
                :FlagTCusto, :FlagAcbMed, :QtdDet, :TempoMedio)
              into :R_COD_REF_DET, :R_DSC_DET, :CustDet;
            R_FK_PRODUTOS = P_FK_PRODUTOS;
            R_SOMA_NIVEL = R_SOMA_NIVEL + CustDet;
          suspend;
        end /* for operacoes_detalhes */
      end /* if Pk_Operacoes */
    end /* for operacoes */
  end /* if P_FkPecas */
end ^

set term ^;
```

grant execute on procedure STP\_GET\_DATA\_COMPONENT to public;

#### **1.4.4.3 Text of procedure STP\_GET\_DATA\_COMPONENT**

```
set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer
)
returns (
  R_FK_PRODUTOS integer,
  R_DSC_PROD varchar(50),
  R_DSC_DET varchar(50),
  R_COD_REF varchar(20),
  R_COD_REF_DET varchar(20),
  R_SOMA_NIVEL double precision
)
as
  declare variable FkTipoDetalhe integer;
  declare variable PkOperacoes integer;
  declare variable FlagTDet smallint;
  declare variable FlagTCusto smallint;
  declare variable FlagAcbMed smallint;
  declare variable TempoMedio double precision;
  declare variable CustDet double precision;
  declare variable QtdDet double precision;
  declare variable PercPerda double precision;
begin
  select FLAG_TCUSTO, FLAG_TACABM
    from PARAMETROS_ESTQ
   where FK_EMPRESAS = :P_FK_EMPRESAS
   into :FlagTCusto, :FlagAcbMed;
```

```

if (:P_FK_PRODUTOS is not null) then
begin
select Pcd.PK_PRODUTOS_CODIGOS, DSC_PROD
  from PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
 where Prd.PK_PRODUTOS = :P_FK_PRODUTOS
    and Pcd.FK_PRODUTOS = Prd.PK_PRODUTOS
    and Pcd.FLAG_TCODE = 0
    and Pcd.FLAG_TBARCODE = 0
 into :R_COD_REF, :R_DSC_PROD;

for select TEMPO_MEDIO, PK_OPERACOES
  from OPERACOES
 where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
 into :TempoMedio, :PkOperacoes do
begin
if (PkOperacoes is not null) then
begin
for select FK_TIPO_DETALHE, FLAG_TDET, QTD_DET,
  PERC_PERDA
  from OPERACOES_DETALHES
 where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
    and FK_OPERACOES = :PkOperacoes
 into :FkTipoDetalhe, :FlagTDet, :QtdDet,
      :PercPerda do
begin
if (FkTipoDetalhe is not null) then
select R_COD_REF, R_DSC_DET, R_CUST_DET
  from STP_CALC_MATERIALS_COST (:FkTipoDetalhe, :FlagTDet,
    :FlagTCusto, :FlagAcbMed, :QtdDet, :TempoMedio)
 into :R_COD_REF_DET, :R_DSC_DET, :CustDet;
R_FK_PRODUTOS = P_FK_PRODUTOS;
R_SOMA_NIVEL = R_SOMA_NIVEL + CustDet;
suspend;
end /* for operacoes_detalhes */
end /* if Pk_Operacoes */
end /* for operacoes */
end /* if P_FkPecas */
end ^

set term ^;

```

#### 1.4.4.4 List of diagrams containing the procedure STP\_GET\_DATA\_COMPONENT

Name	Code
SysTec	SYSTEC

#### 1.4.5 Procedure STP\_GET\_PECAS\_CUSTOS

##### 1.4.5.1 Card of procedure STP\_GET\_PECAS\_CUSTOS

Name	STP_GET_PECAS_CUSTOS
Code	STP_GET_PECAS_CUSTOS
Parent	Package 'SysTec'
Type	Procedure

##### 1.4.5.2 Code preview of procedure STP\_GET\_PECAS\_CUSTOS

```

set term ^;

```

```

create procedure STP_GET_PECAS_CUSTOS (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer
)
returns (
  R_FK_PRODUTOS integer,
  R_DSC_PROD  varchar(50),
  R_DSC_DET   varchar(50),
  R_COD_REF   varchar(20),
  R_COD_REF_DET varchar(20),
  R_SOMA_NIVEL double precision
)
as
  declare variable FkProdutosPecaMnt integer;
  declare variable CustDet      double precision;
begin
  R_SOMA_NIVEL = 0;
  CustDet      = 0;
  select R_FK_PRODUTOS, R_DSC_PROD, R_DSC_DET,
         R_COD_REF, R_COD_REF_DET, R_SOMA_NIVEL
    from STP_GET_DATA_COMPONENT(:P_FK_EMPRESAS, :P_FK_PRODUTOS)
   into :R_FK_PRODUTOS, :R_DSC_PROD, :R_DSC_DET,
        :R_COD_REF, :R_COD_REF_DET, :R_SOMA_NIVEL;
suspend;
for select FK_PRODUTOS_PECAS_MNT
  from PECAS_COMPONENTES
 where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
 into :FkProdutosPecaMnt do
begin
  if (FkProdutosPecaMnt <> R_FK_PRODUTOS) then
  begin
    suspend;
    /* get data from the same procedure with recursive mode */
    for select R_FK_PRODUTOS, R_COD_REF, R_DSC_PROD, R_SOMA_NIVEL
      from STP_GET_PECAS_CUSTOS(:P_FK_EMPRESAS, :FkProdutosPecaMnt)
     into :R_FK_PRODUTOS, :R_COD_REF, :R_DSC_PROD, :R_SOMA_NIVEL do
    begin
      select R_FK_PRODUTOS, R_DSC_PROD, R_DSC_DET,
             R_COD_REF, R_COD_REF_DET, R_SOMA_NIVEL
        from STP_GET_DATA_COMPONENT(:P_FK_EMPRESAS, :R_FK_PRODUTOS)
       into :R_FK_PRODUTOS, :R_DSC_PROD, :R_DSC_DET,
            :R_COD_REF, :R_COD_REF_DET, :R_SOMA_NIVEL;
      suspend;
    end
  end /* if FkPecaMontagem <> P_FkPecas */
end /* for PECAS_COMPONENTES */
end^

set term ^;

```

grant execute on procedure STP\_GET\_PECAS\_CUSTOS to public;

#### **1.4.5.3 Text of procedure STP\_GET\_PECAS\_CUSTOS**

set term ^;

```

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_FK_PRODUTOS integer

```



```

)
returns (
    R_FK_PRODUTOS integer,
    R_DSC_PROD   varchar(50),
    R_DSC_DET    varchar(50),
    R_COD_REF    varchar(20),
    R_COD_REF_DET varchar(20),
    R_SOMA_NIVEL double precision
)
as
declare variable FkProdutosPecaMnt integer;
declare variable CustDet      double precision;
begin
    R_SOMA_NIVEL = 0;
    CustDet      = 0;
    select R_FK_PRODUTOS, R_DSC_PROD, R_DSC_DET,
           R_COD_REF, R_COD_REF_DET, R_SOMA_NIVEL
    from STP_GET_DATA_COMPONENT(:P_FK_EMPRESAS, :P_FK_PRODUTOS)
    into :R_FK_PRODUTOS, :R_DSC_PROD, :R_DSC_DET,
         :R_COD_REF, :R_COD_REF_DET, :R_SOMA_NIVEL;
    suspend;
    for select FK_PRODUTOS_PECAS_MNT
    from PECAS_COMPONENTES
    where FK_PRODUTOS_PECAS = :P_FK_PRODUTOS
    into :FkProdutosPecaMnt do
    begin
        if (FkProdutosPecaMnt <> R_FK_PRODUTOS) then
            begin
                suspend;
                /* get data from the same procedure with recursive mode */
                for select R_FK_PRODUTOS, R_COD_REF, R_DSC_PROD, R_SOMA_NIVEL
                from STP_GET_PECAS_CUSTOS(:P_FK_EMPRESAS, :FkProdutosPecaMnt)
                into :R_FK_PRODUTOS, :R_COD_REF, :R_DSC_PROD, :R_SOMA_NIVEL do
                begin
                    select R_FK_PRODUTOS, R_DSC_PROD, R_DSC_DET,
                           R_COD_REF, R_COD_REF_DET, R_SOMA_NIVEL
                    from STP_GET_DATA_COMPONENT(:P_FK_EMPRESAS, :R_FK_PRODUTOS)
                    into :R_FK_PRODUTOS, :R_DSC_PROD, :R_DSC_DET,
                         :R_COD_REF, :R_COD_REF_DET, :R_SOMA_NIVEL;
                    suspend;
                end
            end /* if FkPecaMontagem <> P_FkPecas */
        end /* for PECAS_COMPONENTES */
    end^

    set term ;^

```

#### 1.4.5.4 List of diagrams containing the procedure STP\_GET\_PECAS\_CUSTOS

Name	Code
SysTec	SYSTEC

#### 1.4.6 Procedure STP\_GET\_TOP\_FATHER

##### 1.4.6.1 Card of procedure STP\_GET\_TOP\_FATHER

Name	STP_GET_TOP_FATHER
Code	STP_GET_TOP_FATHER

<i>Parent</i>	Package 'SysTec'
<i>Type</i>	Procedure

**1.4.6.2 Code preview of procedure STP\_GET\_TOP\_FATHER**

set term ^;

```

create procedure STP_GET_TOP_FATHER (
    P_FK_PRODUTOS_PECAS integer,
    P_MULT          integer
)
RETURNS (
    R_FK_PRODUTOS_PECAS integer,
    R_MAJ_VER          integer,
    R_MIN_VER          integer,
    R_QTD_PEC          float,
    R_COD_REF          varchar(30),
    R_DSC_PROD         varchar(50)
)
as
declare variable FkPecasPesquisa    integer;
declare variable FkPecasPai         integer;
declare variable MajVerPai          integer;
declare variable MinVerPai          integer;
declare variable QtdPecPai          float;
declare variable CodRefPai          varchar(20);
declare variable DscPecPai          varchar(50);
declare variable FkPecasPai2        integer;
declare variable MajVerPai2         integer;
declare variable MinVerPai2         integer;
declare variable QtdPecPai2         float;
declare variable CodRefPai2         varchar(20);
declare variable DscPecPai2         varchar(50);
declare variable TotalFathers        integer;
declare variable TotalGrandFathers  integer;
declare variable AMult               integer;
begin
if ((P_MULT is null) or (P_MULT < 1)) then P_MULT = 1;
TotalFathers = 0;
for select Ppc.FK_PRODUTOS, Ppc.MAJ_VER, Ppc.MIN_VER,
    Pcc.QTD_PEC, Pcd.PK_PRODUTOS_CODIGOS, Prd.DSC_PROD
from PECAS_COMPONENTES Pcc, PRODUTOS_PECAS Ppc,
    PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
where Pcc.FK_PRODUTOS_PECAS <> Pcc.FK_PRODUTOS_PECAS_MNT
and Pcc.FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS
and Ppc.FK_PRODUTOS          = Pcc.FK_PRODUTOS_PECAS
and Prd.PK_PRODUTOS          = Pcc.FK_PRODUTOS_PECAS
and Pcd.FK_PRODUTOS          = Pcc.FK_PRODUTOS_PECAS
into :FkPecasPai, :MajVerPai, :MinVerPai,
    :QtdPecPai, :CodRefPai, :DscPecPai do
begin
TotalGrandFathers = 0;
TotalFathers      = TotalFathers + 1;
FkPecasPesquisa  = FkPecasPai;
AMult = P_MULT * QtdPecPai;
for select R_FK_PRODUTOS_PECAS, R_MAJ_VER, R_MIN_VER, R_QTD_PEC,
    R_COD_REF, R_DSC_PROD
from STP_GET_TOP_FATHER(:FkPecasPesquisa, :AMult)

```

```

into :FkPecasPai2, :MajVerPai2, :MinVerPai2, :QtdPecPai2,
    :CodRefPai2, :DscPecPai2 do
begin
    TotalGrandFathers = TotalGrandFathers + 1;
    R_FK_PRODUTOS_PECAS = FkPecasPai2;
    R_MAJ_VER          = MajVerPai2;
    R_MIN_VER          = MinVerPai2;
    R_QTD_PEC          = P_MULT * QtdPecPai2;
    R_COD_REF          = CodRefPai2;
    R_DSC_PROD         = DscPecPai2;
    suspend;
end
if (TotalGrandFathers < 1) then
begin
    R_FK_PRODUTOS_PECAS = FkPecasPai;
    R_MAJ_VER           = MajVerPai;
    R_MIN_VER           = MinVerPai;
    R_QTD_PEC           = P_MULT * QtdPecPai;
    R_COD_REF           = CodRefPai;
    R_DSC_PROD          = DscPecPai;
    suspend;
end
end
if (TotalFathers < 1) then
begin
    R_FK_PRODUTOS_PECAS = P_FK_PRODUTOS_PECAS;
    select Ppc.MAJ_VER, Ppc.MIN_VER, Pcc.QTD_PEC,
        Pcd.PK_PRODUTOS_CODIGOS, Prd.DSC_PROD
    from PECAS_COMPONENTES Pcc, PRODUTOS_PECAS Ppc,
        PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
    where Pcc.FK_PRODUTOS_PECAS = Pcc.FK_PRODUTOS_PECAS_MNT
        and Pcc.FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS
    into :R_MAJ_VER, :R_MIN_VER, :R_QTD_PEC,
        :R_COD_REF, :R_DSC_PROD;
    if ((R_COD_REF is not null) and (R_QTD_PEC is not null)) then
    begin
        R_QTD_PEC = P_MULT * R_QTD_PEC;
        suspend;
    end
    end
end
end ^

```

set term ;^

grant execute on procedure STP\_GET\_TOP\_FATHER to public;

#### **1.4.6.3 Text of procedure STP\_GET\_TOP\_FATHER**

set term ^;

```

create procedure %PROC% (
    P_FK_PRODUTOS_PECAS integer,
    P_MULT                integer
)
RETURNS (
    R_FK_PRODUTOS_PECAS integer,
    R_MAJ_VER            integer,
    R_MIN_VER            integer,
    R_QTD_PEC            float,

```

```
R_COD_REF      varchar(30),
R_DSC_PROD     varchar(50)
)
as
declare variable FkPecasPesquisa      integer;
declare variable FkPecasPai           integer;
declare variable MajVerPai            integer;
declare variable MinVerPai            integer;
declare variable QtdPecPai            float;
declare variable CodRefPai            varchar(20);
declare variable DscPecPai            varchar(50);
declare variable FkPecasPai2          integer;
declare variable MajVerPai2           integer;
declare variable MinVerPai2           integer;
declare variable QtdPecPai2           float;
declare variable CodRefPai2           varchar(20);
declare variable DscPecPai2           varchar(50);
declare variable TotalFathers          integer;
declare variable TotalGrandFathers    integer;
declare variable AMult                 integer;
begin
if ((P_MULT is null) or (P_MULT < 1)) then P_MULT = 1;
TotalFathers = 0;
for select Ppc.FK_PRODUTOS, Ppc.MAJ_VER, Ppc.MIN_VER,
      Pcc.QTD_PEC, Pcd.PK_PRODUTOS_CODIGOS, Prd.DSC_PROD
from PECAS_COMPONENTES Pcc, PRODUTOS_PECAS Ppc,
      PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
where Pcc.FK_PRODUTOS_PECAS <> Pcc.FK_PRODUTOS_PECAS_MNT
and Pcc.FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS
and Ppc.FK_PRODUTOS = Pcc.FK_PRODUTOS_PECAS
and Prd.PK_PRODUTOS = Pcc.FK_PRODUTOS_PECAS
and Pcd.FK_PRODUTOS = Pcc.FK_PRODUTOS_PECAS
into :FkPecasPai, :MajVerPai, :MinVerPai,
      :QtdPecPai, :CodRefPai, :DscPecPai do
begin
TotalGrandFathers = 0;
TotalFathers = TotalFathers + 1;
FkPecasPesquisa = FkPecasPai;
AMult = P_MULT * QtdPecPai;
for select R_FK_PRODUTOS_PECAS, R_MAJ_VER, R_MIN_VER, R_QTD_PEC,
      R_COD_REF, R_DSC_PROD
from STP_GET_TOP_FATHER(:FkPecasPesquisa, :AMult)
into :FkPecasPai2, :MajVerPai2, :MinVerPai2, :QtdPecPai2,
      :CodRefPai2, :DscPecPai2 do
begin
TotalGrandFathers = TotalGrandFathers + 1;
R_FK_PRODUTOS_PECAS = FkPecasPai2;
R_MAJ_VER = MajVerPai2;
R_MIN_VER = MinVerPai2;
R_QTD_PEC = P_MULT * QtdPecPai2;
R_COD_REF = CodRefPai2;
R_DSC_PROD = DscPecPai2;
suspend;
end
if (TotalGrandFathers < 1) then
begin
R_FK_PRODUTOS_PECAS = FkPecasPai;
R_MAJ_VER = MajVerPai;
R_MIN_VER = MinVerPai;
```

```
R_QTD_PEC      = P_MULT * QtdPecPai;
R_COD_REF      = CodRefPai;
R_DSC_PROD     = DscPecPai;
suspend;
end
end
if (TotalFathers < 1) then
begin
R_FK_PRODUTOS_PECAS = P_FK_PRODUTOS_PECAS;
select Ppc.MAJ_VER, Ppc.MIN_VER, Pcc.QTD_PEC,
      Pcd.PK_PRODUTOS_CODIGOS, Prd.DSC_PROD
from PECAS_COMPONENTES Pcc, PRODUTOS_PECAS Ppc,
      PRODUTOS Prd, PRODUTOS_CODIGOS Pcd
where Pcc.FK_PRODUTOS_PECAS = Pcc.FK_PRODUTOS_PECAS_MNT
and Pcc.FK_PRODUTOS_PECAS_MNT = :P_FK_PRODUTOS_PECAS
into :R_MAJ_VER, :R_MIN_VER, :R_QTD_PEC,
      :R_COD_REF, :R_DSC_PROD;
if ((R_COD_REF is not null) and (R_QTD_PEC is not null)) then
begin
R_QTD_PEC = P_MULT * R_QTD_PEC;
suspend;
end
end
end ^
```

set term ;^

grant execute on procedure %PROC% to public;

#### **1.4.6.4 List of diagrams containing the procedure STP\_GET\_TOP\_FATHER**

<i>Name</i>	<i>Code</i>
SysTec	SYSTEC