

Report

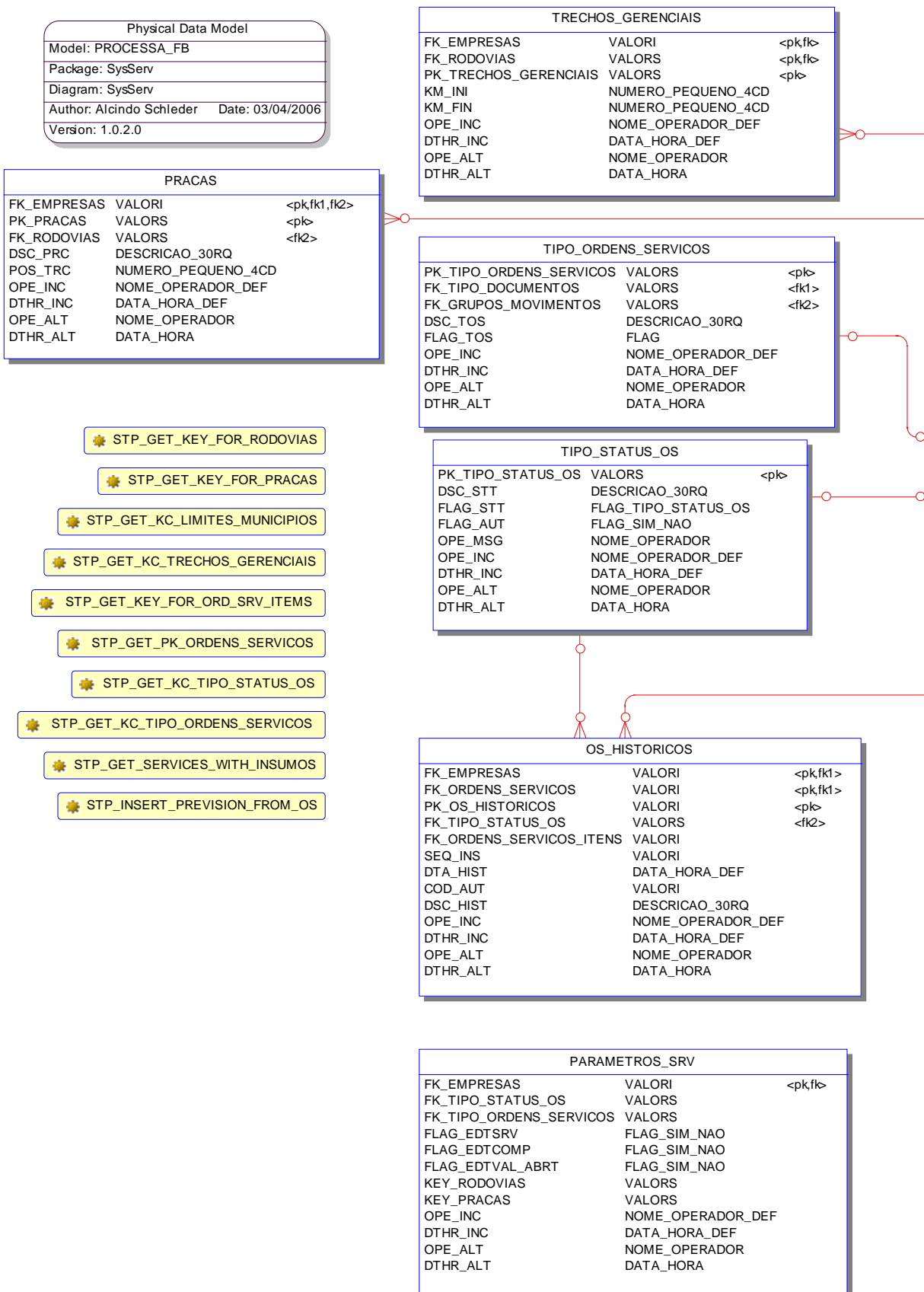
Schleder
04/04/2006

Conteúdo

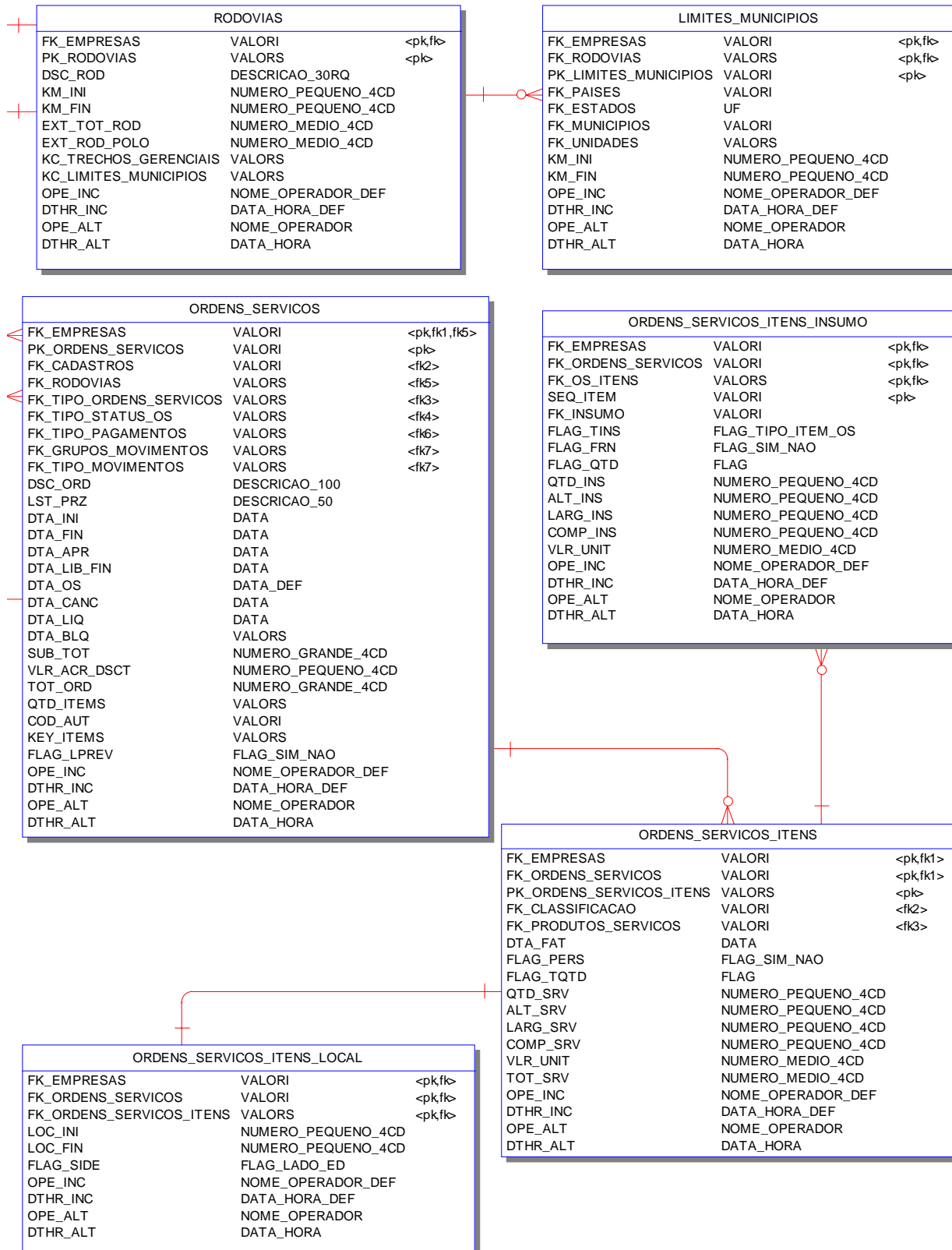
1 Pacote SysServ.....	3
1.1 Diagrama SysServ.....	3
1.1.1 Card of diagram SysServ.....	6
1.2 List of tables.....	6
1.3 Tabelas.....	7
1.3.1 Tabela LIMITES_MUNICIPIOS.....	7
1.3.2 Tabela ORDENS_SERVICOS.....	9
1.3.3 Tabela ORDENS_SERVICOS_ITENS.....	12
1.3.4 Tabela ORDENS_SERVICOS_ITENS_INSUMO.....	15
1.3.5 Tabela ORDENS_SERVICOS_ITENS_LOCAL.....	18
1.3.6 Tabela OS_HISTORICOS.....	20
1.3.7 Tabela PARAMETROS_SRV.....	22
1.3.8 Tabela PRACAS.....	24
1.3.9 Tabela RODOVIAS.....	26
1.3.10 Tabela TIPO_ORDENS_SERVICOS.....	28
1.3.11 Tabela TIPO_STATUS_OS.....	30
1.3.12 Tabela TRECHOS_GERENCIAIS.....	32
1.4 Stored Procedures.....	34
1.4.1 Procedure STP_GET_KC_LIMITES_MUNICIPIOS.....	34
1.4.2 Procedure STP_GET_KC_TIPO_ORDENS_SERVICOS.....	35
1.4.3 Procedure STP_GET_KC_TIPO_STATUS_OS.....	36
1.4.4 Procedure STP_GET_KC_TRECHOS_GERENCIAIS.....	37
1.4.5 Procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS.....	38
1.4.6 Procedure STP_GET_KEY_FOR_PRACAS.....	40
1.4.7 Procedure STP_GET_KEY_FOR_RODOVIAS.....	41
1.4.8 Procedure STP_GET_PK_ORDENS_SERVICOS.....	43
1.4.9 Procedure STP_GET_SERVICES_WITH_INSUMOS.....	45
1.4.10 Procedure STP_INSERT_PREVISION_FROM_OS.....	48

1 Pacote SysServ

1.1 Diagrama SysServ



Módulo de Gerenciamento de Serviços



1.1.1 Card of diagram SysServ

<i>Name</i>	SysServ
<i>Code</i>	SYSSERV
<i>Comment</i>	

1.2 List of tables

Name	Comment
ORDENS_SERVICOS	Tabela que armazena as ordens de serviços
ORDENS_SERVICOS_ITENS	
PRACAS	Tabela que armazena as praças de cobranças dos pedágios
RODOVIAS	Tabela que armazena por rodovias
PARAMETROS_SRV	
LIMITES_MUNICIPIOS	Tabela que armazena os limites dos municípios nas rodovias
TRECHOS_GERENCIAIS	tablea que armazena os trechos gerenciais da rodovia
ORDENS_SERVICOS_ITENS_LOCAL	
TIPO_ORDENS_SERVICOS	Tabela que armazena o tipo de ordem de serviços que podem ser feitas
TIPO_STATUS_OS	Tabela que armazena o Status da Ordem de Serviços
OS_HISTORICOS	Tabela que armazena os Históricos de Status uma OS
ORDENS_SERVICOS_ITENS_INSUMO	Tabela que armazena os insumos, suas quantidade e medidas de um item da ordem de serviços

1.3 Tabelas

1.3.1 Tabela LIMITES_MUNICIPIOS

1.3.1.1 Card of table LIMITES_MUNICIPIOS

Name	LIMITES_MUNICIPIOS
Code	LIMITES_MUNICIPIOS
Comment	Tabela que armazena os limites dos municípios nas rodovias
Primary Key	PK_LIMITES_MUNICIPIOS

1.3.1.2 Lista de Colunas da Tabela LIMITES_MUNICIPIOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	
FK_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
PK_LIMITES_MUNICIPIOS	INTEGER	VALORI	
FK_PAISES	INTEGER	VALORI	Código do País
FK_ESTADOS	CHAR(2)	UF	Código do Estado
FK_MUNICIPIOS	INTEGER	VALORI	Código do Município
FK_UNIDADES	SMALLINT	VALORS	Código da Unida
KM_INI	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
KM_FIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.1.3 Dependências da Tabela LIMITES_MUNICIPIOS

Name	Code	Class Name
FK_LIMITES_MUNIC_RODOVIAS	FK_LIMITES_MUNIC_RODOVIAS	Reference

1.3.1.4 Lista de Chaves da Tabela LIMITES_MUNICIPIOS

Name	Code	Primary
PK_LIMITES_MUNICIPIOS	PK_LIMITES_MUNICIPIOS	X

1.3.1.5 Chave PK_LIMITES_MUNICIPIOS da Tabela LIMITES_MUNICIPIOS

1.3.1.5.1 Lista de Colunas da Chave PK_LIMITES_MUNICIPIOS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_RODOVIAS	FK_RODOVIAS
PK_LIMITES_MUNICIPIOS	PK_LIMITES_MUNICIPIOS

1.3.1.5.2 Dependências da Chave PK_LIMITES_MUNICIPIOS

Name	Code	Class Name
PK_LIMITES_MUNICIPIOS	PK_LIMITES_MUNICIPIOS	Index

1.3.1.6 Lista de Índices LIMITES_MUNICIPIOS

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

1.3.1.7 Índice PK_LIMITES_MUNICIPIOS da Tabela LIMITES_MUNICIPIOS**1.3.1.7.1 Lista de Colunas do Índice PK_LIMITES_MUNICIPIOS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_RODOVIAS	FK_RODOVIAS	Ascending
PK_LIMITES_MUNICIPIOS	PK_LIMITES_MUNICIPIOS	Ascending

1.3.1.8 Lista de Triggers da Tabela LIMITES_MUNICIPIOS

Name	Code
TBIG_LIMITES_MUNICIPIOS	TBIG_LIMITES_MUNICIPIOS
TBU0_LIMITES_MUNICIPIOS	TBU0_LIMITES_MUNICIPIOS

1.3.2 Tabela ORDENS_SERVICOS

1.3.2.1 Card of table ORDENS_SERVICOS

Name	ORDENS_SERVICOS
Code	ORDENS_SERVICOS
Comment	Tabela que armazena as ordens de serviços
Primary Key	PK_ORDENS_SERVICOS

1.3.2.2 Lista de Colunas da Tabela ORDENS_SERVICOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
PK_ORDENS_SERVICOS	INTEGER	VALORI	
FK_CADASTROS	INTEGER	VALORI	Código do Cadastro
FK_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
FK_TIPO_ORDENS_SERVICOS	SMALLINT	VALORS	Código da Ordem de Serviços
FK_TIPO_STATUS_OS	SMALLINT	VALORS	Código do Status da OS
FK_TIPO_PAGAMENTOS	SMALLINT	VALORS	Código do Tipo de Pagamento
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Grupo de Movimentos
FK_TIPO_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
DSC_ORD	VARCHAR(100)	DESCRICAO_100	Descrição da Ordem de Serviços
LST_PRZ	VARCHAR(50)	DESCRICAO_50	Lista de prazos da condição de pagamento
DTA_INI	DATE	DATA	Data de previsão Início do Serviço
DTA_FIN	DATE	DATA	Data prevista para finalização do Serviço
DTA_APR	DATE	DATA	Data da aprovação da OS
DTA_LIB_FIN	DATE	DATA	Data da Liberação do Financeiro
DTA_OS	DATE	DATA_DEF	Data da Ordem de Serviços
DTA_CANC	DATE	DATA	Data do Cancelamento
DTA_LIQ	DATE	DATA	Data de Liquidação da Ordem de Serviços
DTA_BLK	SMALLINT	VALORS	Data do Bloqueio
SUB_TOT	NUMERIC(18,4)	NUMERO_GRANDE_4CD	SubTotal da Ordem de Serviços
VLR_ACR_DSC	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Valor de Acréscimo / Desconto Acréscimo ==> Valor Positivo Desconto ==> Valor Negativo

Name	Data Type	Domain	Comment
TOT_ORD	NUMERIC(18,4)	NUMERO_GRADE_4CD	Total da Ordem de serviços
QTD_ITEMS	SMALLINT	VALORS	Quantidade de Ítems
COD_AUT	INTEGER	VALORI	Código da última autorização...
KEY_ITEMS	SMALLINT	VALORS	controle da chave do ítem
FLAG_LPRE V	SMALLINT	FLAG_SIM_NAO	Flag que indica se o lançamento do Previsto já foi feito
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.2.3 Dependências da Tabela ORDENS_SERVICOS

Name	Code	Class Name
FK_ORDENS_SERVIC_ORDSRV_I TENS	FK_ORDENS_SERVIC_ORDSRV_I TENS	Reference
FK_OS_HISTORICOS_ORDENS_S ERVIC	FK_OS_HISTORICOS_ORDENS_S ERVIC	Reference
FK_ORDENS_SERVIC_EMPRESA S	FK_ORDENS_SERVIC_EMPRESA S	Reference
FK_ORDENS_SERVIC_CADASTR OS	FK_ORDENS_SERVIC_CADASTR OS	Reference
FK_TIPO_ORDENS_S_TIPO_DOC UMENT	FK_TIPO_ORDENS_S_TIPO_DOC UMENT	Reference
FK_ORDENS_SERVIC_TIPO_STAT US_O	FK_ORDENS_SERVIC_TIPO_STAT US_O	Reference
FK_ORDENS_SERVIC_RODOVIAS	FK_ORDENS_SERVIC_RODOVIAS	Reference
FK_ORDENS_SERVIC_TIPO_PAG AMENT	FK_ORDENS_SERVIC_TIPO_PAG AMENT	Reference
FK_ORDENS_SERVIC_TIPO_MOVI MENT	FK_ORDENS_SERVIC_TIPO_MOVI MENT	Reference

1.3.2.4 Lista de Chaves da Tabela ORDENS_SERVICOS

Name	Code	Primary
PK_ORDENS_SERVICOS	PK_ORDENS_SERVICOS	X

1.3.2.5 Chave PK_ORDENS_SERVICOS da Tabela ORDENS_SERVICOS

1.3.2.5.1 Lista de Colunas da Chave PK_ORDENS_SERVICOS

Name	Code
PK_ORDENS_SERVICOS	PK_ORDENS_SERVICOS

Name	Code
FK_EMPRESAS	FK_EMPRESAS

1.3.2.5.2 Dependências da Chave PK_ORDENS_SERVICOS

Name	Code	Class Name
PK_ORDENS_SERVICOS	PK_ORDENS_SERVICOS	Index
FK_ORDENS_SERVIC_ORDSRV_I TENS	FK_ORDENS_SERVIC_ORDSRV_I TENS	Reference
FK_OS_HISTORICOS_ORDENS_S ERVIC	FK_OS_HISTORICOS_ORDENS_S ERVIC	Reference

1.3.2.6 Lista de Índices ORDENS_SERVICOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_ORDENS_ SERVICOS	PK_ORDENS_ SERVICOS	X		X		
IDX_ORDENS_ SERVICOS001	IDX_ORDENS_ SERVICOS001					

1.3.2.7 Índice PK_ORDENS_SERVICOS da Tabela ORDENS_SERVICOS

1.3.2.7.1 Lista de Colunas do Índice PK_ORDENS_SERVICOS

Name	Code	Sort
PK_ORDENS_SERVICOS	PK_ORDENS_SERVICOS	Ascending
FK_EMPRESAS	FK_EMPRESAS	Ascending

1.3.2.8 Índice IDX_ORDENS_SERVICOS001 da Tabela ORDENS_SERVICOS

1.3.2.8.1 Lista de Colunas do Índice IDX_ORDENS_SERVICOS001

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
DTA_OS	DTA_OS	Ascending
PK_ORDENS_SERVICOS	PK_ORDENS_SERVICOS	Ascending

1.3.2.9 Lista de Triggers da Tabela ORDENS_SERVICOS

Name	Code
TBIG_ORDENS_SERVICOS	TBIG_ORDENS_SERVICOS
TBU0_ORDENS_SERVICOS	TBU0_ORDENS_SERVICOS

1.3.3 Tabela ORDENS_SERVICOS_ITENS

1.3.3.1 Card of table ORDENS_SERVICOS_ITENS

Name	ORDENS_SERVICOS_ITENS
Code	ORDENS_SERVICOS_ITENS
Comment	
Primary Key	PK_ORDENS_SERVICOS_ITENS

1.3.3.2 Lista de Colunas da Tabela ORDENS_SERVICOS_ITENS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_ORDENS_SERVICOS	INTEGER	VALORI	
PK_ORDENS_SERVICOS_ITENS	SMALLINT	VALORS	
FK_CLASSIFICACAO	INTEGER	VALORI	Código do Plano de Contas (Sequencial com DV)
FK_PRODUTOS_SERVICOS	INTEGER	VALORI	Código do serviço
DTA_FAT	DATE	DATA	Data do Faturamento
FLAG_PERS	SMALLINT	FLAG_SIM_NAO	Flag que indica que o item tem itens personalizados
FLAG_TQTD	SMALLINT	FLAG	Flag que que armazena o modo de entrada da quantidade Obs: Na linguagem deve-se passar o valor a um ponteiro e depois transformar o valor apontado do ponteiro em um tipo enumerado pré definido
QTD_SRV	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Quantidade de Serviços contratados
ALT_SRV	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Altura do Serviço
LARG_SRV	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Largura do Serviço
COMP_SRV	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Comprimento
VLR_UNIT	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Valor Unitário do Serviço
TOT_SRV	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Valor Total dos Serviços Contratos
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.3.3 Dependências da Tabela ORDENS_SERVICOS_ITENS

Name	Code	Class Name
FK_ORDENS_SERVICOS_ORDENS_LOCAL	FK_ORDENS_SERVICOS_ORDENS_LOCAL	Reference
FK_OS_ITENS_OS_ITENS_INSUMOS	FK_OS_ITENS_OS_ITENS_INSUMOS	Reference
FK_ORDENS_SERVIC_ORDSRV_ITENS	FK_ORDENS_SERVIC_ORDSRV_ITENS	Reference
FK_ORDENS_SERVIC_CLASSIFICACAO	FK_ORDENS_SERVIC_CLASSIFICACAO	Reference
FK_ORDENS_SERVIC_PRODUTOS_SERV	FK_ORDENS_SERVIC_PRODUTOS_SERV	Reference

1.3.3.4 Lista de Chaves da Tabela ORDENS_SERVICOS_ITENS

Name	Code	Primary
PK_ORDENS_SERVICOS_ITENS	PK_ORDENS_SERVICOS_ITENS	X

1.3.3.5 Chave PK_ORDENS_SERVICOS_ITENS da Tabela ORDENS_SERVICOS_ITENS

1.3.3.5.1 Lista de Colunas da Chave PK_ORDENS_SERVICOS_ITENS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS
PK_ORDENS_SERVICOS_ITENS	PK_ORDENS_SERVICOS_ITENS

1.3.3.5.2 Dependências da Chave PK_ORDENS_SERVICOS_ITENS

Name	Code	Class Name
PK_ORDENS_SERVICOS_ITENS	PK_ORDENS_SERVICOS_ITENS	Index
FK_ORDENS_SERVICOS_ORDENS_LOCAL	FK_ORDENS_SERVICOS_ORDENS_LOCAL	Reference
FK_OS_ITENS_OS_ITENS_INSUMOS	FK_OS_ITENS_OS_ITENS_INSUMOS	Reference

1.3.3.6 Lista de Índices ORDENS_SERVICOS_ITENS

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

1.3.3.7 Índice PK_ORDENS_SERVICOS_ITENS da Tabela ORDENS_SERVICOS_ITENS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS	Ascending
PK_ORDENS_SERVICOS_ITENS	PK_ORDENS_SERVICOS_ITENS	Ascending

1.3.3.8 Lista de Triggers da Tabela ORDENS_SERVICOS_ITENS

Name	Code
TBIG_ORDENS_SERVICOS_ITENS	TBIG_ORDENS_SERVICOS_ITENS
TBU0_ORDENS_SERVICOS_ITENS	TBU0_ORDENS_SERVICOS_ITENS

1.3.4 Tabela ORDENS_SERVICOS_ITENS_INSUMO

1.3.4.1 Card of table ORDENS_SERVICOS_ITENS_INSUMO

Name	ORDENS_SERVICOS_ITENS_INSUMO
Code	ORDENS_SERVICOS_ITENS_INSUMO
Comment	Tabela que armazena os insumos, suas quantidade e medidas de um ítem da ordem de serviços
Primary Key	PK_ORDENS_SERVICOS_ITENS_INSUM

1.3.4.2 Lista de Colunas da Tabela ORDENS_SERVICOS_ITENS_INSUMO

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_ORDENS_SERVICOS	INTEGER	VALORI	
FK_OS_ITENS	SMALLINT	VALORS	
SEQ_ITEM	INTEGER	VALORI	Número da sequência de aplicação do serviço
FK_INSUMO	INTEGER	VALORI	Código do Insumo
FLAG_TINS	SMALLINT	FLAG_TIPO_ITEM_OS	Flag que indica o tipo de insumo utilizado numa ordem de serviços 0 ==> Serviço (TIPO_COMPOSICOES) 1 ==> Atividades (SERVICOS_IND) 2 ==> Material (INSUMOS)
FLAG_FRN	SMALLINT	FLAG_SIM_NAO	Flag que indica se o material usado é fornecido ou não. No caso de OS de Entrada ==> indica que o usuário forneceu o material ou não No caso de OS de Saída ==> indica que o cliente forneceu o material ou não
FLAG_QTD	SMALLINT	FLAG	Flag que que armazena o modo de entrada da quantidade Obs: Na linguagem deve-se passar o valor a um ponteiro e depois transformar o valor apontado do ponteiro em um tipo enumerado pré definido
QTD_INS	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
ALT_INS	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Altura do Serviço
LARG_INS	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Largura do Serviço
COMP_INS	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Comprimento

Name	Data Type	Domain	Comment
VLR_UNIT	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Campo que armazena o valor unitário de cada insumo... pode ser que um serviço possa ter que somar os insumos para ter o valor do serviço
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.4.3 Dependências da Tabela ORDENS_SERVICOS_ITENS_INSUMO

Name	Code	Class Name
FK_OS_ITENS_OS_ITENS_INSUM OS	FK_OS_ITENS_OS_ITENS_INSUM OS	Reference

1.3.4.4 Lista de Chaves da Tabela ORDENS_SERVICOS_ITENS_INSUMO

Name	Code	Primary
PK_ORDENS_SERVICOS_ITENS_I NSUM	PK_ORDENS_SERVICOS_ITENS_I NSUM	X

1.3.4.5 Chave PK_ORDENS_SERVICOS_ITENS_INSUM da Tabela ORDENS_SERVICOS_ITENS_INSUMO

1.3.4.5.1 Lista de Colunas da Chave PK_ORDENS_SERVICOS_ITENS_INSUM

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS
FK_OS_ITENS	FK_OS_ITENS
SEQ_ITEM	SEQ_ITEM

1.3.4.5.2 Dependências da Chave PK_ORDENS_SERVICOS_ITENS_INSUM

Name	Code	Class Name
PK_ORDENS_SERVICOS_ITENS_I NSUM	PK_ORDENS_SERVICOS_ITENS_I NSUM	Index

1.3.4.6 Lista de Índices ORDENS_SERVICOS_ITENS_INSUMO

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

**1.3.4.7 Índice PK_ORDENS_SERVICOS_ITENS_INSUM da Tabela
ORDENS_SERVICOS_ITENS_INSUMO****1.3.4.7.1 Lista de Colunas do Índice PK_ORDENS_SERVICOS_ITENS_INSUM**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS	Ascending
FK_OS_ITENS	FK_OS_ITENS	Ascending
SEQ_ITEM	SEQ_ITEM	Ascending

1.3.4.8 Lista de Triggers da Tabela ORDENS_SERVICOS_ITENS_INSUMO

Name	Code
TBU0_ORDENS_SERVICOS_ITENS_INS	TBU0_ORDENS_SERVICOS_ITENS_INS

1.3.5 Tabela ORDENS_SERVICOS_ITENS_LOCAL

1.3.5.1 Card of table ORDENS_SERVICOS_ITENS_LOCAL

Name	ORDENS_SERVICOS_ITENS_LOCAL
Code	ORDENS_SERVICOS_ITENS_LOCAL
Comment	
Primary Key	PK_ORDENS_SERVICOS_ITENS_LOCAL

1.3.5.2 Lista de Colunas da Tabela ORDENS_SERVICOS_ITENS_LOCAL

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_ORDENS_SERVICOS	INTEGER	VALORI	
FK_ORDENS_SERVICOS_ITENS	SMALLINT	VALORS	
LOC_INI	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Localização do Marco inicial
LOC_FIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Localização do marco final
FLAG_SIDE	SMALLINT	FLAG_LADO_ED	Flag que indica qual lado pertence o registro 0 ==> Lado Esquerdo 1 ==> Lado Direito 2 ==> Ambos
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.5.3 Dependências da Tabela ORDENS_SERVICOS_ITENS_LOCAL

Name	Code	Class Name
FK_ORDENS_SERVICOS_ORDENS_LOCAL	FK_ORDENS_SERVICOS_ORDENS_LOCAL	Reference

1.3.5.4 Lista de Chaves da Tabela ORDENS_SERVICOS_ITENS_LOCAL

Name	Code	Primary
PK_ORDENS_SERVICOS_ITENS_LOCAL	PK_ORDENS_SERVICOS_ITENS_LOCAL	X

1.3.5.5 Chave PK_ORDENS_SERVICOS_ITENS_LOCAL da Tabela ORDENS_SERVICOS_ITENS_LOCAL

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS
FK_ORDENS_SERVICOS_ITENS	FK_ORDENS_SERVICOS_ITENS

1.3.5.5.2 Dependências da Chave PK_ORDENS_SERVICOS_ITENS_LOCAL

Name	Code	Class Name
PK_ORDENS_SERVICOS_ITENS_LOCAL	PK_ORDENS_SERVICOS_ITENS_LOCAL	Index

1.3.5.6 Lista de Índices ORDENS_SERVICOS_ITENS_LOCAL

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

1.3.5.7 Índice PK_ORDENS_SERVICOS_ITENS_LOCAL da Tabela ORDENS_SERVICOS_ITENS_LOCAL**1.3.5.7.1 Lista de Colunas do Índice PK_ORDENS_SERVICOS_ITENS_LOCAL**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS	Ascending
FK_ORDENS_SERVICOS_ITENS	FK_ORDENS_SERVICOS_ITENS	Ascending

1.3.5.8 Lista de Triggers da Tabela ORDENS_SERVICOS_ITENS_LOCAL

Name	Code
TBU0_ORDENS_SERVICOS_ITENS_LOCA	TBU0_ORDENS_SERVICOS_ITENS_LOCA

1.3.6 Tabela OS_HISTORICOS

1.3.6.1 Card of table OS_HISTORICOS

Name	OS_HISTORICOS
Code	OS_HISTORICOS
Comment	Tabela que armazena os Históricos de Status uma OS
Primary Key	PK_OS_HISTORICOS

1.3.6.2 Lista de Colunas da Tabela OS_HISTORICOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_ORDENS_SERVICOS	INTEGER	VALORI	
PK_OS_HISTORICOS	INTEGER	VALORI	Data e Hora do Histórico
FK_TIPO_STATUS_OS	SMALLINT	VALORS	Código do Status da OS
FK_ORDENS_SERVICOS_ITENS	INTEGER	VALORI	Campo que armazena um vínculo com o item para gerar o históricos dos itens
SEQ_INS	INTEGER	VALORI	Campo que armazena o valor da sequência do insumo, podendo gerar históricos para cada insumo do item (depende de FK_ORDENS_SERVICOS_ITENS)
DTA_HIST	TIMESTAMP	DATA_HORA_DEF	Data e hora do histórico
COD_AUT	INTEGER	VALORI	Código de Autorização
DSC_HIST	VARCHAR(30)	DESCRICAO_30RQ	Descrição do histórico
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.6.3 Dependências da Tabela OS_HISTORICOS

Name	Code	Class Name
FK_OS_HISTORICOS_ORDENS_SERVIC	FK_OS_HISTORICOS_ORDENS_SERVIC	Reference
FK_OS_HISTORICOS_TIPO_STATUS_O	FK_OS_HISTORICOS_TIPO_STATUS_O	Reference

1.3.6.4 Lista de Chaves da Tabela OS_HISTORICOS

Name	Code	Primary
------	------	---------

Name	Code	Primary
PK_OS_HISTORICOS	PK_OS_HISTORICOS	X

1.3.6.5 Chave PK_OS_HISTORICOS da Tabela OS_HISTORICOS

1.3.6.5.1 Lista de Colunas da Chave PK_OS_HISTORICOS

Name	Code
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS
FK_EMPRESAS	FK_EMPRESAS
PK_OS_HISTORICOS	PK_OS_HISTORICOS

1.3.6.5.2 Dependências da Chave PK_OS_HISTORICOS

Name	Code	Class Name
PK_OS_HISTORICOS	PK_OS_HISTORICOS	Index

1.3.6.6 Lista de Índices OS_HISTORICOS

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

1.3.6.7 Índice PK_OS_HISTORICOS da Tabela OS_HISTORICOS

1.3.6.7.1 Lista de Colunas do Índice PK_OS_HISTORICOS

Name	Code	Sort
FK_ORDENS_SERVICOS	FK_ORDENS_SERVICOS	Ascending
FK_EMPRESAS	FK_EMPRESAS	Ascending
PK_OS_HISTORICOS	PK_OS_HISTORICOS	Ascending

1.3.6.8 Lista de Triggers da Tabela OS_HISTORICOS

Name	Code
TBU0_OS_HISTORICOS	TBU0_OS_HISTORICOS

1.3.7 Tabela PARAMETROS_SRV

1.3.7.1 Card of table PARAMETROS_SRV

Name	PARAMETROS_SRV
Code	PARAMETROS_SRV
Comment	
Primary Key	PK_PARAMETROS_SRV

1.3.7.2 Lista de Colunas da Tabela PARAMETROS_SRV

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	
FK_TIPO_STATUS_OS	SMALLINT	VALORS	Status default de inserção da OS
FK_TIPO_ORDENSERVICOS	SMALLINT	VALORS	Código do tipo da ordem de produção default na inserção de novas OSs
FLAG_EDTSRV	SMALLINT	FLAG_SIM_NAO	Flag que indica se o Serviço pode ser editado após sua gravação
FLAG_EDTCOMP	SMALLINT	FLAG_SIM_NAO	Flag que indica se pode ser editada a composição de um serviço ou não
FLAG_EDTVAL_ABRT	SMALLINT	FLAG_SIM_NAO	Flag que indica se o valor da OS pode ser alterado somente com o status de abertura
KEY_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
KEY_PRAÇAS	SMALLINT	VALORS	Controlador da chave da tabela de praças
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.7.3 Dependências da Tabela PARAMETROS_SRV

Name	Code	Class Name
FK_PARAMETROS_SRV_EMPRESAS	FK_PARAMETROS_SRV_EMPRESAS	Reference

1.3.7.4 Lista de Chaves da Tabela PARAMETROS_SRV

Name	Code	Primary
PK_PARAMETROS_SRV	PK_PARAMETROS_SRV	X

1.3.7.5 Chave PK_PARAMETROS_SRV da Tabela PARAMETROS_SRV

1.3.7.5.1 Lista de Colunas da Chave PK_PARAMETROS_SRV

Name	Code
FK_EMPRESAS	FK_EMPRESAS

1.3.7.5.2 Dependências da Chave PK_PARAMETROS_SRV

Name	Code	Class Name
PK_PARAMETROS_SRV	PK_PARAMETROS_SRV	Index

1.3.7.6 Lista de Índices PARAMETROS_SRV

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

1.3.7.7 Índice PK_PARAMETROS_SRV da Tabela PARAMETROS_SRV**1.3.7.7.1 Lista de Colunas do Índice PK_PARAMETROS_SRV**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending

1.3.7.8 Lista de Triggers da Tabela PARAMETROS_SRV

Name	Code
TBU0_PARAMETROS_SRV	TBU0_PARAMETROS_SRV

1.3.8 Tabela PRACAS

1.3.8.1 Card of table PRACAS

Name	PRACAS
Code	PRACAS
Comment	Tabela que armazena as praças de cobranças dos pedágios
Primary Key	PK_PRACAS

1.3.8.2 Lista de Colunas da Tabela PRACAS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
PK_PRACAS	SMALLINT	VALORS	
FK_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
DSC_PRC	VARCHAR(30)	DESCRICAO_30RQ	Descrição da Praça
POS_TRC	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Kilômetro da Praça de Pedágio na rodovia
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.8.3 Dependências da Tabela PRACAS

Name	Code	Class Name
FK_CATEGORIAS_PRODUTOS_PRACAS	FK_CATEGORIAS_PRODUTOS_PRACAS	Reference
FK_PRACAS_PISTAS_PRACAS	FK_PRACAS_PISTAS_PRACAS	Reference
FK_PRACAS_OPERADERES_PRACAS	FK_PRACAS_OPERADERES_PRACAS	Reference
FK_VINCULO_TOCOR_PRACAS_PRACAS	FK_VINCULO_TOCOR_PRACAS_PRACAS	Reference
FK_PRACAS_EMPRESAS	FK_PRACAS_EMPRESAS	Reference
FK_PRACAS_RODOVIAS	FK_PRACAS_RODOVIAS	Reference
PRACAS	PRACAS	Shortcut of table

1.3.8.4 Lista de Chaves da Tabela PRACAS

Name	Code	Primary
PK_PRACAS	PK_PRACAS	X

1.3.8.5 Chave PK_PRACAS da Tabela PRACAS

1.3.8.5.1 Lista de Colunas da Chave PK_PRACAS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
PK_PRACAS	PK_PRACAS

1.3.8.5.2 Dependências da Chave PK_PRACAS

Name	Code	Class Name
PK_PRACAS	PK_PRACAS	Index
FK_CATEGORIAS_PRODUTOS_P RACAS	FK_CATEGORIAS_PRODUTOS_P RACAS	Reference
FK_PRACAS_PISTAS_PRACAS	FK_PRACAS_PISTAS_PRACAS	Reference
FK_PRACAS_OPERADERES_PRA CAS	FK_PRACAS_OPERADERES_PRA CAS	Reference
FK_VINCULO_TOCOR_PRACAS_P RACAS	FK_VINCULO_TOCOR_PRACAS_P RACAS	Reference

1.3.8.6 Lista de Índices PRACAS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_PRACAS	PK_PRACAS	X		X		

1.3.8.7 Índice PK_PRACAS da Tabela PRACAS**1.3.8.7.1 Lista de Colunas do Índice PK_PRACAS**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
PK_PRACAS	PK_PRACAS	Ascending

1.3.8.8 Lista de Triggers da Tabela PRACAS

Name	Code
TBIG_PRACAS	TBIG_PRACAS
TBU0_PRACAS	TBU0_PRACAS

1.3.9 Tabela RODOVIAS

1.3.9.1 Card of table RODOVIAS

Name	RODOVIAS
Code	RODOVIAS
Comment	Tabela que armazena por rodovias
Primary Key	PK_RODOVIAS

1.3.9.2 Lista de Colunas da Tabela RODOVIAS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	
PK_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
DSC_ROD	VARCHAR(30)	DESCRICAO_30RQ	Descrição da rodovia
KM_INI	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Marco Inicial da concessão na Rodovia
KM_FIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Marco Final da concessão na Rodovia
EXT_TOT_ROD	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Extensão total da Rodovia
EXT_ROD_POLO	NUMERIC(11,4)	NUMERO_MEDIO_4CD	Extensão da rodovia pertencente ao pólo
KC_TRECHOS_GERENCIAIS	SMALLINT	VALORS	Key Control da tabela `Trechos Gerenciais
KC_LIMITES_MUNICIPIOS	SMALLINT	VALORS	Key Control da tabela de limites de municípios
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.9.3 Dependências da Tabela RODOVIAS

Name	Code	Class Name
FK_LIMITES_MUNIC_RODOVIAS	FK_LIMITES_MUNIC_RODOVIAS	Reference
FK_PRACAS_RODOVIAS	FK_PRACAS_RODOVIAS	Reference
FK_TRECHOS_GEREN_RODOVIAS	FK_TRECHOS_GEREN_RODOVIAS	Reference
FK_ORDENS_SERVIC_RODOVIAS	FK_ORDENS_SERVIC_RODOVIAS	Reference
FK_RODOVIAS_EMPRESAS	FK_RODOVIAS_EMPRESAS	Reference

Name	Code	Primary
PK_RODOVIAS	PK_RODOVIAS	X

1.3.9.5 Chave PK_RODOVIAS da Tabela RODOVIAS

1.3.9.5.1 Lista de Colunas da Chave PK_RODOVIAS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
PK_RODOVIAS	PK_RODOVIAS

1.3.9.5.2 Dependências da Chave PK_RODOVIAS

Name	Code	Class Name
PK_RODOVIAS	PK_RODOVIAS	Index
FK_LIMITES_MUNIC_RODOVIAS	FK_LIMITES_MUNIC_RODOVIAS	Reference
FK_PRACAS_RODOVIAS	FK_PRACAS_RODOVIAS	Reference
FK_TRECHOS_GEREN_RODOVIAS	FK_TRECHOS_GEREN_RODOVIAS	Reference
FK_ORDENS_SERVIC_RODOVIAS	FK_ORDENS_SERVIC_RODOVIAS	Reference

1.3.9.6 Lista de Índices RODOVIAS

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

1.3.9.7 Índice PK_RODOVIAS da Tabela RODOVIAS

1.3.9.7.1 Lista de Colunas do Índice PK_RODOVIAS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
PK_RODOVIAS	PK_RODOVIAS	Ascending

1.3.9.8 Lista de Triggers da Tabela RODOVIAS

Name	Code
TBIG_RODOVIAS	TBIG_RODOVIAS
TBU0_RODOVIAS	TBU0_RODOVIAS

1.3.10 Tabela TIPO_ORDENS_SERVICOS

1.3.10.1 Card of table TIPO_ORDENS_SERVICOS

Name	TIPO_ORDENS_SERVICOS
Code	TIPO_ORDENS_SERVICOS
Comment	Tabela que armazena o tipo de ordem de serviços que podem ser feitas
Primary Key	PK_TIPO_ORDENS_SERVICOS

1.3.10.2 Lista de Colunas da Tabela TIPO_ORDENS_SERVICOS

Name	Data Type	Domain	Comment
PK_TIPO_ORDENS_SERVICOS	SMALLINT	VALORS	Código da Ordem de Serviços
FK_TIPO_DOCUMENTOS	SMALLINT	VALORS	Tipo de documento vinculado ao Tipo de OS
FK_GRUPOS_MOVIMENTOS	SMALLINT	VALORS	Código do Tipo de Movimento
DSC_TOS	VARCHAR(30)	DESCRICAO_30RQ	Descrição do tipo de ordens de serviços
FLAG_TOS	SMALLINT	FLAG	Flag que indica o tipo de Ordem de Serviço será confeccionada 0 ==> Medições (topografia) 1 ==> Oficina Mecânica 2 ==> Construção Civil 3 ==> Tecnologia (Informática)
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.10.3 Dependências da Tabela TIPO_ORDENS_SERVICOS

Name	Code	Class Name
FK_TIPO_ORDENS_S_TIPO_DOCUMENT	FK_TIPO_ORDENS_S_TIPO_DOCUMENT	Reference
FK_TIPO_ORDENS_S_TIPO_DOCUMENT	FK_TIPO_ORDENS_S_TIPO_DOCUMENT	Reference
FK_TIPO_ORDENS_S_GRUPOS_MOVIME	FK_TIPO_ORDENS_S_GRUPOS_MOVIME	Reference

1.3.10.4 Lista de Chaves da Tabela TIPO_ORDENS_SERVICOS

Name	Code	Primary
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS	X

1.3.10.5 Chave PK_TIPO_ORDENS_SERVICOS da Tabela TIPO_ORDENS_SERVICOS**1.3.10.5.1 Lista de Colunas da Chave PK_TIPO_ORDENS_SERVICOS**

Name	Code
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS

1.3.10.5.2 Dependências da Chave PK_TIPO_ORDENS_SERVICOS

Name	Code	Class Name
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS	Index
FK_TIPO_ORDENS_S_TIPO_DOCUMENT	FK_TIPO_ORDENS_S_TIPO_DOCUMENT	Reference

1.3.10.6 Lista de Índices TIPO_ORDENS_SERVICOS

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternate Key
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS	X		X		
IDX_TIPO_ORDENS_SERVICOS_001	IDX_TIPO_ORDENS_SERVICOS_001					

1.3.10.7 Índice PK_TIPO_ORDENS_SERVICOS da Tabela TIPO_ORDENS_SERVICOS**1.3.10.7.1 Lista de Colunas do Índice PK_TIPO_ORDENS_SERVICOS**

Name	Code	Sort
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS	Ascending

1.3.10.8 Índice IDX_TIPO_ORDENS_SERVICOS_001 da Tabela TIPO_ORDENS_SERVICOS**1.3.10.8.1 Lista de Colunas do Índice IDX_TIPO_ORDENS_SERVICOS_001**

Name	Code	Sort
FK_GRUPOS_MOVIMENTOS	FK_GRUPOS_MOVIMENTOS	Ascending
PK_TIPO_ORDENS_SERVICOS	PK_TIPO_ORDENS_SERVICOS	Ascending

1.3.10.9 Lista de Triggers da Tabela TIPO_ORDENS_SERVICOS

Name	Code
TBIG_TIPO_ORDENS_SERVICOS	TBIG_TIPO_ORDENS_SERVICOS
TBU0_TIPO_ORDENS_SERVICOS	TBU0_TIPO_ORDENS_SERVICOS

1.3.11 Tabela TIPO_STATUS_OS

1.3.11.1 Card of table TIPO_STATUS_OS

Name	TIPO_STATUS_OS
Code	TIPO_STATUS_OS
Comment	Tabela que armazena o Status da Ordem de Serviços
Primary Key	PK_TIPO_STATUS_OS

1.3.11.2 Lista de Colunas da Tabela TIPO_STATUS_OS

Name	Data Type	Domain	Comment
PK_TIPO_ST ATUS_OS	SMALLINT	VALORS	Código do Status da OS
DSC_STT	VARCHAR(30)	DESCRICAO_30RQ	Descrição do Status
FLAG_STT	SMALLINT	FLAG_TIPO_STATUS_OS	Flag que indica a situação da OS 0 ==> Aberta 1 ==> Faturamento Parcial 2 ==> Faturamento Total / Liquidação 3 ==> Cancelamento 4 ==> Bloqueio 5 ==> Aprovada
FLAG_AUT	SMALLINT	FLAG_SIM_NAO	Flag que indica se o status necessita autorização
OPE_MSG	VARCHAR(10)	NOME_OPERADOR	Nome do operador para enviar mensagem de alerta
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.11.3 Dependências da Tabela TIPO_STATUS_OS

Name	Code	Class Name
FK_ORDENS_SERVIC_TIPO_STAT US_O	FK_ORDENS_SERVIC_TIPO_STAT US_O	Reference
FK_OS_HISTORICOS_TIPO_STAT US_O	FK_OS_HISTORICOS_TIPO_STAT US_O	Reference

1.3.11.4 Lista de Chaves da Tabela TIPO_STATUS_OS

Name	Code	Primary
PK_TIPO_STATUS_OS	PK_TIPO_STATUS_OS	X

1.3.11.5 Chave PK_TIPO_STATUS_OS da Tabela TIPO_STATUS_OS

Name	Code
PK_TIPO_STATUS_OS	PK_TIPO_STATUS_OS

1.3.11.5.2 Dependências da Chave PK_TIPO_STATUS_OS

Name	Code	Class Name
PK_TIPO_STATUS_OS	PK_TIPO_STATUS_OS	Index
FK_ORDENS_SERVIC_TIPO_STAT US_O	FK_ORDENS_SERVIC_TIPO_STAT US_O	Reference
FK_OS_HISTORICOS_TIPO_STAT US_O	FK_OS_HISTORICOS_TIPO_STAT US_O	Reference

1.3.11.6 Lista de Índices TIPO_STATUS_OS

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

1.3.11.7 Índice PK_TIPO_STATUS_OS da Tabela TIPO_STATUS_OS**1.3.11.7.1 Lista de Colunas do Índice PK_TIPO_STATUS_OS**

Name	Code	Sort
PK_TIPO_STATUS_OS	PK_TIPO_STATUS_OS	Ascending

1.3.11.8 Lista de Triggers da Tabela TIPO_STATUS_OS

Name	Code
TBIG_TIPO_STATUS_OS	TBIG_TIPO_STATUS_OS
TBU0_TIPO_STATUS_OS	TBU0_TIPO_STATUS_OS

1.3.12 Tabela TRECHOS_GERENCIAIS

1.3.12.1 Card of table TRECHOS_GERENCIAIS

Name	TRECHOS_GERENCIAIS
Code	TRECHOS_GERENCIAIS
Comment	tablea que armazena os trechos gerenciais da rodovia
Primary Key	PK_TRECHOS_GERENCIAIS

1.3.12.2 Lista de Colunas da Tabela TRECHOS_GERENCIAIS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	
FK_RODOVIAS	SMALLINT	VALORS	Código da Rodovia
PK_TRECHOS_GERENCIAIS	SMALLINT	VALORS	
KM_INI	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
KM_FIN	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	
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	Operador que fez a última alteração no registro
DTHR_ALT	TIMESTAMP	DATA_HORA	Data e Hora da última alteração feita no registro

1.3.12.3 Dependências da Tabela TRECHOS_GERENCIAIS

Name	Code	Class Name
FK_TRECHOS_GEREN_RODOVIAS	FK_TRECHOS_GEREN_RODOVIAS	Reference

1.3.12.4 Lista de Chaves da Tabela TRECHOS_GERENCIAIS

Name	Code	Primary
PK_TRECHOS_GERENCIAIS	PK_TRECHOS_GERENCIAIS	X

1.3.12.5 Chave PK_TRECHOS_GERENCIAIS da Tabela TRECHOS_GERENCIAIS

1.3.12.5.1 Lista de Colunas da Chave PK_TRECHOS_GERENCIAIS

Name	Code
PK_TRECHOS_GERENCIAIS	PK_TRECHOS_GERENCIAIS
FK_RODOVIAS	FK_RODOVIAS
FK_EMPRESAS	FK_EMPRESAS

1.3.12.5.2 Dependências da Chave PK_TRECHOS_GERENCIAIS

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

Name	Code	Class Name
PK_TRECHOS_GERENCIAIS	PK_TRECHOS_GERENCIAIS	Index

1.3.12.6 Lista de Índices TRECHOS_GERENCIAIS

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

1.3.12.7 Índice PK_TRECHOS_GERENCIAIS da Tabela TRECHOS_GERENCIAIS**1.3.12.7.1 Lista de Colunas do Índice PK_TRECHOS_GERENCIAIS**

Name	Code	Sort
PK_TRECHOS_GERENCIAIS	PK_TRECHOS_GERENCIAIS	Ascending
FK_RODOVIAS	FK_RODOVIAS	Ascending
FK_EMPRESAS	FK_EMPRESAS	Ascending

1.3.12.8 Lista de Triggers da Tabela TRECHOS_GERENCIAIS

Name	Code
TBIG_TRECHOS_GERENCIAIS	TBIG_TRECHOS_GERENCIAIS
TBU0_TRECHOS_GERENCIAIS	TBU0_TRECHOS_GERENCIAIS

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

<i>Name</i>	STP_GET_KC_LIMITES_MUNICIPIOS
<i>Code</i>	STP_GET_KC_LIMITES_MUNICIPIOS
<i>Parent</i>	Package 'SysServ'
<i>Type</i>	Procedure

1.4.1.2 Code preview of procedure STP_GET_KC_LIMITES_MUNICIPIOS

set term ^;

```

create procedure STP_GET_KC_LIMITES_MUNICIPIOS (
  P_FK_EMPRESAS integer,
  P_FK_RODOVIAS smallint
)
returns (
  R_PK_LIMITES_MUNICIPIOS smallint
)
as
begin
  R_PK_LIMITES_MUNICIPIOS = -1;
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
    ( P_FK_RODOVIAS is not null) and (P_FK_RODOVIAS > 0)) then
  begin
    select KC_LIMITES_MUNICIPIOS
      from RODOVIAS
     where FK_EMPRESAS = :P_FK_EMPRESAS
       and PK_RODOVIAS = :P_FK_RODOVIAS
    into :R_PK_LIMITES_MUNICIPIOS;
    if ((R_PK_LIMITES_MUNICIPIOS is not null) and
      ( R_PK_LIMITES_MUNICIPIOS >= 0)) then
    begin
      R_PK_LIMITES_MUNICIPIOS = R_PK_LIMITES_MUNICIPIOS + 1;
      update RODOVIAS set
        KC_LIMITES_MUNICIPIOS = :R_PK_LIMITES_MUNICIPIOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
        and PK_RODOVIAS = :P_FK_RODOVIAS;
    end
  end
  suspend;
end ^

```

set term ^;

grant execute on procedure STP_GET_KC_LIMITES_MUNICIPIOS to public;

1.4.1.3 Text of procedure STP_GET_KC_LIMITES_MUNICIPIOS

set term ^;

```

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_FK_RODOVIAS smallint
)

```

```
returns (  
  R_PK_LIMITES_MUNICIPIOS smallint  
)  
as  
begin  
  R_PK_LIMITES_MUNICIPIOS = -1;  
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and  
    ( P_FK_RODOVIAS is not null) and (P_FK_RODOVIAS > 0)) then  
    begin  
      select KC_LIMITES_MUNICIPIOS  
        from RODOVIAS  
       where FK_EMPRESAS = :P_FK_EMPRESAS  
         and PK_RODOVIAS = :P_FK_RODOVIAS  
         into :R_PK_LIMITES_MUNICIPIOS;  
      if ((R_PK_LIMITES_MUNICIPIOS is not null) and  
        ( R_PK_LIMITES_MUNICIPIOS >= 0)) then  
        begin  
          R_PK_LIMITES_MUNICIPIOS = R_PK_LIMITES_MUNICIPIOS + 1;  
          update RODOVIAS set  
            KC_LIMITES_MUNICIPIOS = :R_PK_LIMITES_MUNICIPIOS  
           where FK_EMPRESAS = :P_FK_EMPRESAS  
             and PK_RODOVIAS = :P_FK_RODOVIAS;  
        end  
      end  
    end  
    suspend;  
  end ^  
  
set term ;^
```

1.4.1.4 List of diagrams containing the procedure STP_GET_KC_LIMITES_MUNICIPIOS

Name	Code
SysServ	SYSSERV

1.4.2 Procedure STP_GET_KC_TIPO_ORDENS_SERVICOS

1.4.2.1 Card of procedure STP_GET_KC_TIPO_ORDENS_SERVICOS

Name	STP_GET_KC_TIPO_ORDENS_SERVICOS
Code	STP_GET_KC_TIPO_ORDENS_SERVICOS
Parent	Package 'SysServ'
Type	Procedure

1.4.2.2 Code preview of procedure STP_GET_KC_TIPO_ORDENS_SERVICOS

```
set term ^;  
  
create procedure STP_GET_KC_TIPO_ORDENS_SERVICOS  
returns (  
  R_PK_TIPO_ORDENS_SERVICOS smallint  
)  
as  
begin  
  R_PK_TIPO_ORDENS_SERVICOS = Gen_Id(TIPO_ORDENS_SERVICOS, 1);  
end ^  
  
set term ;^;
```

grant execute on procedure STP_GET_KC_TIPO_ORDENS_SERVICOS to public;

1.4.2.3 Text of procedure STP_GET_KC_TIPO_ORDENS_SERVICOS

```
set term ^;

create procedure %PROC%
returns (
  R_PK_TIPO_ORDENS_SERVICOS smallint
)
as
begin
  R_PK_TIPO_ORDENS_SERVICOS = Gen_Id(TIPO_ORDENS_SERVICOS, 1);
end ^

set term ;^
```

1.4.2.4 List of diagrams containing the procedure STP_GET_KC_TIPO_ORDENS_SERVICOS

Name	Code
SysServ	SYSSERV

1.4.3 Procedure STP_GET_KC_TIPO_STATUS_OS

1.4.3.1 Card of procedure STP_GET_KC_TIPO_STATUS_OS

Name	STP_GET_KC_TIPO_STATUS_OS
Code	STP_GET_KC_TIPO_STATUS_OS
Parent	Package 'SysServ'
Type	Procedure

1.4.3.2 Code preview of procedure STP_GET_KC_TIPO_STATUS_OS

```
set term ^;

create procedure STP_GET_KC_TIPO_STATUS_OS
returns (
  R_PK_TIPO_STATUS_OS smallint
)
as
begin
  R_PK_TIPO_STATUS_OS = Gen_Id(TIPO_STATUS_OS, 1);
end ^

set term ;^;

grant execute on procedure STP_GET_KC_TIPO_STATUS_OS to public;
```

1.4.3.3 Text of procedure STP_GET_KC_TIPO_STATUS_OS

```
set term ^;

create procedure %PROC%
returns (
  R_PK_TIPO_STATUS_OS smallint
)
as
begin
  R_PK_TIPO_STATUS_OS = Gen_Id(TIPO_STATUS_OS, 1);
end ^
```

set term ;^

1.4.3.4 List of diagrams containing the procedure STP_GET_KC_TIPO_STATUS_OS

<i>Name</i>	<i>Code</i>
SysServ	SYSSERV

1.4.4 Procedure STP_GET_KC_TRECHOS_GERENCIAIS

1.4.4.1 Card of procedure STP_GET_KC_TRECHOS_GERENCIAIS

<i>Name</i>	STP_GET_KC_TRECHOS_GERENCIAIS
<i>Code</i>	STP_GET_KC_TRECHOS_GERENCIAIS
<i>Parent</i>	Package 'SysServ'
<i>Type</i>	Procedure

1.4.4.2 Code preview of procedure STP_GET_KC_TRECHOS_GERENCIAIS

set term ^;

```
create procedure STP_GET_KC_TRECHOS_GERENCIAIS (
  P_FK_EMPRESAS integer,
  P_FK_RODOVIAS smallint
)
returns (
  R_PK_TRECHOS_GERENCIAIS smallint
)
as
begin
  R_PK_TRECHOS_GERENCIAIS = -1;
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
    ( P_FK_RODOVIAS is not null) and (P_FK_RODOVIAS > 0)) then
  begin
    select KC_TRECHOS_GERENCIAIS
      from RODOVIAS
     where FK_EMPRESAS = :P_FK_EMPRESAS
       and PK_RODOVIAS = :P_FK_RODOVIAS
    into :R_PK_TRECHOS_GERENCIAIS;
    if ((R_PK_TRECHOS_GERENCIAIS is not null) and
      ( R_PK_TRECHOS_GERENCIAIS >= 0)) then
    begin
      R_PK_TRECHOS_GERENCIAIS = R_PK_TRECHOS_GERENCIAIS + 1;
      update RODOVIAS set
        KC_TRECHOS_GERENCIAIS = :R_PK_TRECHOS_GERENCIAIS
       where FK_EMPRESAS = :P_FK_EMPRESAS
         and PK_RODOVIAS = :P_FK_RODOVIAS;
    end
  end
  suspend;
end ^
```

set term ;^;

grant execute on procedure STP_GET_KC_TRECHOS_GERENCIAIS to public;

1.4.4.3 Text of procedure STP_GET_KC_TRECHOS_GERENCIAIS

set term ^;

```
create procedure %PROC% (  
  P_FK_EMPRESAS integer,  
  P_FK_RODOVIAS smallint  
)  
returns (  
  R_PK_TRECHOS_GERENCIAIS smallint  
)  
as  
begin  
  R_PK_TRECHOS_GERENCIAIS = -1;  
  if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and  
    ( P_FK_RODOVIAS is not null) and (P_FK_RODOVIAS > 0)) then  
  begin  
    select KC_TRECHOS_GERENCIAIS  
      from RODOVIAS  
    where FK_EMPRESAS = :P_FK_EMPRESAS  
      and PK_RODOVIAS = :P_FK_RODOVIAS  
    into :R_PK_TRECHOS_GERENCIAIS;  
    if ((R_PK_TRECHOS_GERENCIAIS is not null) and  
      ( R_PK_TRECHOS_GERENCIAIS >= 0)) then  
    begin  
      R_PK_TRECHOS_GERENCIAIS = R_PK_TRECHOS_GERENCIAIS + 1;  
      update RODOVIAS set  
        KC_TRECHOS_GERENCIAIS = :R_PK_TRECHOS_GERENCIAIS  
      where FK_EMPRESAS = :P_FK_EMPRESAS  
        and PK_RODOVIAS = :P_FK_RODOVIAS;  
    end  
  end  
  suspend;  
end ^  
  
set term ^;
```

1.4.4.4 List of diagrams containing the procedure STP_GET_KC_TRECHOS_GERENCIAIS

Name	Code
SysServ	SYSSERV

1.4.5 Procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS

1.4.5.1 Card of procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS

Name	STP_GET_KEY_FOR_ORD_SRV_ITEMS
Code	STP_GET_KEY_FOR_ORD_SRV_ITEMS
Parent	Package 'SysServ'
Type	Procedure

1.4.5.2 Code preview of procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS

```
set term ^;  
  
create procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS (  
  P_FK_EMPRESAS integer,  
  P_PK_ORDENS_SERVICOS integer  
)  
returns (  
  R_PK_ORDENS_SERVICOS_ITEMS smallint
```

```

)
as
begin
  R_PK_ORDENS_SERVICOS_ITEMS = -1;
  if (((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) and
      ((P_PK_ORDENS_SERVICOS is not null) or (P_PK_ORDENS_SERVICOS > 0))) then
    begin
      select KEY_ITEMS
      from ORDENS_SERVICOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
      and PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
      into :R_PK_ORDENS_SERVICOS_ITEMS;
    if (R_PK_ORDENS_SERVICOS_ITEMS is null) then
      R_PK_ORDENS_SERVICOS_ITEMS = -1;
    if (R_PK_ORDENS_SERVICOS_ITEMS >= 0) then
      begin
        R_PK_ORDENS_SERVICOS_ITEMS = R_PK_ORDENS_SERVICOS_ITEMS + 1;
        update ORDENS_SERVICOS set
          KEY_ITEMS = :R_PK_ORDENS_SERVICOS_ITEMS
          where FK_EMPRESAS = :P_FK_EMPRESAS
          and PK_ORDENS_SERVICOS = :R_PK_ORDENS_SERVICOS_ITEMS;
      end
    end
  suspend;
end ^

set term ^;

```

grant execute on procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS to public;

1.4.5.3 Text of procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS

```

set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS integer,
  P_PK_ORDENS_SERVICOS integer
)
returns (
  R_PK_ORDENS_SERVICOS_ITEMS smallint
)
as
begin
  R_PK_ORDENS_SERVICOS_ITEMS = -1;
  if (((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) and
      ((P_PK_ORDENS_SERVICOS is not null) or (P_PK_ORDENS_SERVICOS > 0))) then
    begin
      select KEY_ITEMS
      from ORDENS_SERVICOS
      where FK_EMPRESAS = :P_FK_EMPRESAS
      and PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
      into :R_PK_ORDENS_SERVICOS_ITEMS;
    if (R_PK_ORDENS_SERVICOS_ITEMS is null) then
      R_PK_ORDENS_SERVICOS_ITEMS = -1;
    if (R_PK_ORDENS_SERVICOS_ITEMS >= 0) then
      begin
        R_PK_ORDENS_SERVICOS_ITEMS = R_PK_ORDENS_SERVICOS_ITEMS + 1;
        update ORDENS_SERVICOS set
          KEY_ITEMS = :R_PK_ORDENS_SERVICOS_ITEMS

```

```

where FK_EMPRESAS      = :P_FK_EMPRESAS
and PK_ORDENS_SERVICOS = :R_PK_ORDENS_SERVICOS_ITEMS;
end
end
suspend;
end ^

```

set term ;^

1.4.5.4 List of diagrams containing the procedure STP_GET_KEY_FOR_ORD_SRV_ITEMS

<i>Name</i>	<i>Code</i>
SysServ	SYSSERV

1.4.6 Procedure STP_GET_KEY_FOR_PRACAS

1.4.6.1 Card of procedure STP_GET_KEY_FOR_PRACAS

<i>Name</i>	STP_GET_KEY_FOR_PRACAS
<i>Code</i>	STP_GET_KEY_FOR_PRACAS
<i>Parent</i>	Package 'SysServ'
<i>Type</i>	Procedure

1.4.6.2 Code preview of procedure STP_GET_KEY_FOR_PRACAS

set term ^;

```

create procedure STP_GET_KEY_FOR_PRACAS (
  P_FK_EMPRESAS integer
)
returns (
  R_PK_PRACAS smallint
)
as
begin
  R_PK_PRACAS = -1;
  if ((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) then
  begin
    select KEY_PRACAS
    from PARAMETROS_SRV
    where FK_EMPRESAS = :P_FK_EMPRESAS
    into :R_PK_PRACAS;
    if ((R_PK_PRACAS is null) or (R_PK_PRACAS = -1)) then
    begin
      insert into PARAMETROS_SRV
      (FK_EMPRESAS, FK_TIPO_STATUS_OS, FK_TIPO_ORDENS_SERVICOS,
      FLAG_EDTSRV, FLAG_EDTCOMP, FLAG_EDTVAL_ABRT, KEY_RODOVIAS,
      KEY_PRACAS)
      values
      (:P_FK_EMPRESAS, null, null, 0, 0, 1, 0, 0);
      R_PK_PRACAS = 0;
    end
    R_PK_PRACAS = R_PK_PRACAS + 1;
    update PARAMETROS_SRV set
      KEY_PRACAS = :R_PK_PRACAS
    where FK_EMPRESAS = :P_FK_EMPRESAS;
  end
  suspend;

```


end ^

set term ^;

grant execute on procedure STP_GET_KEY_FOR_PRACAS to public;

1.4.6.3 Text of procedure STP_GET_KEY_FOR_PRACAS

set term ^;

```
create procedure %PROC% (  
  P_FK_EMPRESAS integer  
)  
returns (  
  R_PK_PRACAS smallint  
)  
as  
begin  
  R_PK_PRACAS = -1;  
  if ((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) then  
  begin  
    select KEY_PRACAS  
      from PARAMETROS_SRV  
     where FK_EMPRESAS = :P_FK_EMPRESAS  
     into :R_PK_PRACAS;  
    if ((R_PK_PRACAS is null) or (R_PK_PRACAS = -1)) then  
    begin  
      insert into PARAMETROS_SRV  
        (FK_EMPRESAS, FK_TIPO_STATUS_OS, FK_TIPO_ORDENS_SERVICOS,  
         FLAG_EDTSRV, FLAG_EDTCOMP, FLAG_EDTVAL_ABRT, KEY_RODOVIAS,  
         KEY_PRACAS)  
      values  
        (:P_FK_EMPRESAS, null, null, 0, 0, 1, 0, 0);  
      R_PK_PRACAS = 0;  
    end  
    R_PK_PRACAS = R_PK_PRACAS + 1;  
    update PARAMETROS_SRV set  
      KEY_PRACAS = :R_PK_PRACAS  
     where FK_EMPRESAS = :P_FK_EMPRESAS;  
  end  
suspend;  
end ^
```

set term ^;

1.4.6.4 List of diagrams containing the procedure STP_GET_KEY_FOR_PRACAS

<i>Name</i>	<i>Code</i>
SysServ	SYSSERV

1.4.7 Procedure STP_GET_KEY_FOR_RODOVIAS

1.4.7.1 Card of procedure STP_GET_KEY_FOR_RODOVIAS

<i>Name</i>	STP_GET_KEY_FOR_RODOVIAS
<i>Code</i>	STP_GET_KEY_FOR_RODOVIAS
<i>Parent</i>	Package 'SysServ'
<i>Type</i>	Procedure

1.4.7.2 Code preview of procedure STP_GET_KEY_FOR_RODOVIAS

```
set term ^;

create procedure STP_GET_KEY_FOR_RODOVIAS (
  P_FK_EMPRESAS integer
)
returns (
  R_PK_RODOVIAS smallint
)
as
begin
  R_PK_RODOVIAS = -1;
  if ((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) then
  begin
    select KEY_RODOVIAS
    from PARAMETROS_SRV
    where FK_EMPRESAS = :P_FK_EMPRESAS
    into :R_PK_RODOVIAS;
    if ((R_PK_RODOVIAS is null) or (R_PK_RODOVIAS = -1)) then
    begin
      insert into PARAMETROS_SRV
      (FK_EMPRESAS, FK_TIPO_STATUS_OS, FK_TIPO_ORDENS_SERVICOS,
       FLAG_EDTSRV, FLAG_EDTCOMP, FLAG_EDTVAL_ABRT, KEY_RODOVIAS,
       KEY_PRACAS)
      values
      (:P_FK_EMPRESAS, null, null, 0, 0, 1, 0, 0);
      R_PK_RODOVIAS = 0;
    end
    R_PK_RODOVIAS = R_PK_RODOVIAS + 1;
    update PARAMETROS_SRV set
      KEY_RODOVIAS = :R_PK_RODOVIAS
    where FK_EMPRESAS = :P_FK_EMPRESAS;
  end
  suspend;
end ^

set term ^;
```

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

1.4.7.3 Text of procedure STP_GET_KEY_FOR_RODOVIAS

```
set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS integer
)
returns (
  R_PK_RODOVIAS smallint
)
as
begin
  R_PK_RODOVIAS = -1;
  if ((P_FK_EMPRESAS is not null) or (P_FK_EMPRESAS > 0)) then
  begin
    select KEY_RODOVIAS
    from PARAMETROS_SRV
    where FK_EMPRESAS = :P_FK_EMPRESAS
```

```
into :R_PK_RODOVIAS;
if ((R_PK_RODOVIAS is null) or (R_PK_RODOVIAS = -1)) then
begin
insert into PARAMETROS_SRV
(FK_EMPRESAS, FK_TIPO_STATUS_OS, FK_TIPO_ORDENS_SERVICOS,
FLAG_EDTSRV, FLAG_EDTCOMP, FLAG_EDTVAL_ABRT, KEY_RODOVIAS,
KEY_PRACAS)
values
(:P_FK_EMPRESAS, null, null, 0, 0, 1, 0, 0);
R_PK_RODOVIAS = 0;
end
R_PK_RODOVIAS = R_PK_RODOVIAS + 1;
update PARAMETROS_SRV set
KEY_RODOVIAS = :R_PK_RODOVIAS
where FK_EMPRESAS = :P_FK_EMPRESAS;
end
suspend;
end ^

set term ^;
```

1.4.7.4 List of diagrams containing the procedure STP_GET_KEY_FOR_RODOVIAS

<i>Name</i>	<i>Code</i>
SysServ	SYSSERV

1.4.8 Procedure STP_GET_PK_ORDENS_SERVICOS

1.4.8.1 Card of procedure STP_GET_PK_ORDENS_SERVICOS

<i>Name</i>	STP_GET_PK_ORDENS_SERVICOS
<i>Code</i>	STP_GET_PK_ORDENS_SERVICOS
<i>Parent</i>	Package 'SysServ'
<i>Type</i>	Procedure

1.4.8.2 Code preview of procedure STP_GET_PK_ORDENS_SERVICOS

```
set term ^;

create procedure STP_GET_PK_ORDENS_SERVICOS (
P_FK_EMPRESAS integer,
P_FK_TIPO_DOCUMENTOS integer
)
returns (
R_PK_ORDENS_SERVICOS smallint
)
as
begin
R_PK_ORDENS_SERVICOS = null;
if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
(P_FK_TIPO_DOCUMENTOS is not null) and (P_FK_TIPO_DOCUMENTOS > 0)) then
begin
select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_DOCUMENTOS
into :R_PK_ORDENS_SERVICOS;
if (R_PK_ORDENS_SERVICOS is null) then
begin
```

```
insert into TIPO_DOCUMENTOS_NUMERACAO
(FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)
values
(:P_FK_EMPRESAS, :P_FK_TIPO_DOCUMENTOS, 0);
R_PK_ORDENS_SERVICOS = 0;
end
R_PK_ORDENS_SERVICOS = R_PK_ORDENS_SERVICOS + 1;
update TIPO_DOCUMENTOS_NUMERACAO set
NUM_DOC = :R_PK_ORDENS_SERVICOS
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_DOCUMENTOS;
end
suspend;
end ^

set term ;^

grant execute on procedure STP_GET_PK_ORDENS_SERVICOS to public;
```

1.4.8.3 Text of procedure STP_GET_PK_ORDENS_SERVICOS

```
set term ^;

create procedure %PROC% (
P_FK_EMPRESAS integer,
P_FK_TIPO_DOCUMENTOS integer
)
returns (
R_PK_ORDENS_SERVICOS smallint
)
as
begin
R_PK_ORDENS_SERVICOS = null;
if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
(P_FK_TIPO_DOCUMENTOS is not null) and (P_FK_TIPO_DOCUMENTOS > 0)) then
begin
select NUM_DOC from TIPO_DOCUMENTOS_NUMERACAO
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_DOCUMENTOS
into :R_PK_ORDENS_SERVICOS;
if (R_PK_ORDENS_SERVICOS is null) then
begin
insert into TIPO_DOCUMENTOS_NUMERACAO
(FK_EMPRESAS, FK_TIPO_DOCUMENTOS, NUM_DOC)
values
(:P_FK_EMPRESAS, :P_FK_TIPO_DOCUMENTOS, 0);
R_PK_ORDENS_SERVICOS = 0;
end
R_PK_ORDENS_SERVICOS = R_PK_ORDENS_SERVICOS + 1;
update TIPO_DOCUMENTOS_NUMERACAO set
NUM_DOC = :R_PK_ORDENS_SERVICOS
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_DOCUMENTOS;
end
suspend;
end ^

set term ;^
```

grant execute on procedure %PROC% to public;

1.4.8.4 List of diagrams containing the procedure STP_GET_PK_ORDENS_SERVICOS

Name	Code
SysServ	SYSSERV

1.4.9 Procedure STP_GET_SERVICES_WITH_INSUMOS

1.4.9.1 Card of procedure STP_GET_SERVICES_WITH_INSUMOS

Name	STP_GET_SERVICES_WITH_INSUMOS
Code	STP_GET_SERVICES_WITH_INSUMOS
Parent	Package 'SysServ'
Type	Procedure

1.4.9.2 Code preview of procedure STP_GET_SERVICES_WITH_INSUMOS

set term ^;

```
create procedure STP_GET_SERVICES_WITH_INSUMOS
returns (
  R_PK_PRODUTOS    integer,
  R_FK_INSUMOS     integer,
  R_FK_UNIDADES    integer,
  R_COD_REF        varchar(30),
  R_FK_CLASSIFICACAO integer,
  R_DSC_PROD       varchar(60),
  R_FLAG_QTD       smallint,
  R_FLAG_ALT       smallint,
  R_FLAG_LARG      smallint,
  R_FLAG_COMP      smallint,
  R_VLR_UNIT       numeric(9, 4),
  R_QTD_INS        numeric(9, 4),
  R_CFC_CONV       numeric(9, 4),
  R_SEQ_INS        integer,
  R_FLAG_TINS      smallint,
  R_FLAG_DEF       smallint,
  R_FLAG_TUNI      smallint,
  R_FLAG_FUNI      smallint,
  R_MED_DEF        numeric(9,4)
)
as
begin
  for select PK_PRODUTOS, COD_REF, FK_CLASSIFICACAO,
    DSC_PROD, FLAG_QTD, FLAG_ALT, FLAG_LARG,
    FLAG_COMP, QTD_UNI, VLR_UNIT, FLAG_TUNI,
    FLAG_FUNI, PK_UNIDADES
  from VW_SERVICES
  order by DSC_PROD
  into :R_PK_PRODUTOS, :R_COD_REF, :R_FK_CLASSIFICACAO,
    :R_DSC_PROD, :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
    :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT, :R_FLAG_TUNI,
    :R_FLAG_FUNI, :R_FK_UNIDADES do
  begin
    R_FLAG_TINS = -1;
```

```

suspend;
for select FLAG_TCOMP, FK_INSUMOS, QTD_PROD, SEQ_COMP, FLAG_DEF,
  MED_DEF
  from PRODUTOS_COMPOSICOES
  where FK_PRODUTOS = :R_PK_PRODUTOS
  order by SEQ_COMP
  into :R_FLAG_TINS, :R_FK_INSUMOS, :R_QTD_INS, :R_SEQ_INS,
       :R_FLAG_DEF, :R_MED_DEF do
begin
  R_PK_PRODUTOS = 0;
  R_DSC_PROD   = "";
  R_COD_REF    = "";
  R_FLAG_QTD   = 0;
  R_FLAG_ALT   = 0;
  R_FLAG_LARG  = 0;
  R_FLAG_COMP  = 0;
  R_CFC_CONV   = 0.0;
  R_VLR_UNIT   = 0.0;
  if (R_FLAG_TINS = 0) then
  begin
    select PK_PRODUTOS, DSC_PROD, COD_REF, FLAG_QTD,
      FLAG_ALT, FLAG_LARG, FLAG_COMP,
      QTD_UNI, VLR_UNIT, PK_UNIDADES
    from VW_ATIVIDADES
    where PK_PRODUTOS = :R_FK_INSUMOS
    into :R_PK_PRODUTOS, :R_DSC_PROD, :R_COD_REF,
         :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
         :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT,
         :R_FK_UNIDADES;
  end
  if (R_FLAG_TINS = 1) then
  begin
    select PK_PRODUTOS, DSC_PROD, COD_REF, FLAG_QTD,
      FLAG_ALT, FLAG_LARG, FLAG_COMP,
      QTD_UNI, VLR_UNIT, PK_UNIDADES
    from VW_MATERIAL
    where PK_PRODUTOS = :R_FK_INSUMOS
    into :R_PK_PRODUTOS, :R_DSC_PROD, :R_COD_REF,
         :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
         :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT,
         :R_FK_UNIDADES;
  end
  suspend;
end
end
end ^

set term ;^

```

grant execute on procedure STP_GET_SERVICES_WITH_INSUMOS to public;

1.4.9.3 Text of procedure STP_GET_SERVICES_WITH_INSUMOS

```

set term ^;

create procedure %PROC%
returns (
  R_PK_PRODUTOS   integer,

```

```

R_FK_INSUMOS      integer,
R_FK_UNIDADES     integer,
R_COD_REF         varchar(30),
R_FK_CLASSIFICACAO integer,
R_DSC_PROD        varchar(60),
R_FLAG_QTD        smallint,
R_FLAG_ALT        smallint,
R_FLAG_LARG       smallint,
R_FLAG_COMP       smallint,
R_VLR_UNIT        numeric(9, 4),
R_QTD_INS         numeric(9, 4),
R_CFC_CONV        numeric(9, 4),
R_SEQ_INS         integer,
R_FLAG_TINS       smallint,
R_FLAG_DEF        smallint,
R_FLAG_TUNI       smallint,
R_FLAG_FUNI       smallint,
R_MED_DEF         numeric(9,4)
)
as
begin
  for select PK_PRODUTOS, COD_REF, FK_CLASSIFICACAO,
    DSC_PROD, FLAG_QTD, FLAG_ALT, FLAG_LARG,
    FLAG_COMP, QTD_UNI, VLR_UNIT, FLAG_TUNI,
    FLAG_FUNI, PK_UNIDADES
  from VW_SERVICES
  order by DSC_PROD
  into :R_PK_PRODUTOS, :R_COD_REF, :R_FK_CLASSIFICACAO,
    :R_DSC_PROD, :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
    :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT, :R_FLAG_TUNI,
    :R_FLAG_FUNI, :R_FK_UNIDADES do
begin
  R_FLAG_TINS = -1;
  suspend;
  for select FLAG_TCOMP, FK_INSUMOS, QTD_PROD, SEQ_COMP, FLAG_DEF,
    MED_DEF
  from PRODUTOS_COMPOSICOES
  where FK_PRODUTOS = :R_PK_PRODUTOS
  order by SEQ_COMP
  into :R_FLAG_TINS, :R_FK_INSUMOS, :R_QTD_INS, :R_SEQ_INS,
    :R_FLAG_DEF, :R_MED_DEF do
begin
  R_PK_PRODUTOS = 0;
  R_DSC_PROD  = "";
  R_COD_REF   = "";
  R_FLAG_QTD  = 0;
  R_FLAG_ALT  = 0;
  R_FLAG_LARG = 0;
  R_FLAG_COMP = 0;
  R_CFC_CONV  = 0.0;
  R_VLR_UNIT  = 0.0;
  if (R_FLAG_TINS = 0) then
  begin
    select PK_PRODUTOS, DSC_PROD, COD_REF, FLAG_QTD,
      FLAG_ALT, FLAG_LARG, FLAG_COMP,
      QTD_UNI, VLR_UNIT, PK_UNIDADES
    from VW_ATIVIDADES
    where PK_PRODUTOS = :R_FK_INSUMOS

```

```

into :R_PK_PRODUTOS, :R_DSC_PROD, :R_COD_REF,
      :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
      :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT,
      :R_FK_UNIDADES;
end
if (R_FLAG_TINS = 1) then
begin
select PK_PRODUTOS, DSC_PROD, COD_REF, FLAG_QTD,
      FLAG_ALT, FLAG_LARG, FLAG_COMP,
      QTD_UNI, VLR_UNIT, PK_UNIDADES
from VW_MATERIAL
where PK_PRODUTOS = :R_FK_INSUMOS
into :R_PK_PRODUTOS, :R_DSC_PROD, :R_COD_REF,
      :R_FLAG_QTD, :R_FLAG_ALT, :R_FLAG_LARG,
      :R_FLAG_COMP, :R_CFC_CONV, :R_VLR_UNIT,
      :R_FK_UNIDADES;
end
suspend;
end
end
end ^

set term ;^

grant execute on procedure %PROC% to public;

```

1.4.9.4 List of diagrams containing the procedure STP_GET_SERVICES_WITH_INSUMOS

Name	Code
SysServ	SYSSERV

1.4.10 Procedure STP_INSERT_PREVISION_FROM_OS

1.4.10.1 Card of procedure STP_INSERT_PREVISION_FROM_OS

Name	STP_INSERT_PREVISION_FROM_OS
Code	STP_INSERT_PREVISION_FROM_OS
Parent	Package 'SysServ'
Type	Procedure

1.4.10.2 Code preview of procedure STP_INSERT_PREVISION_FROM_OS

```

set term ^;

create procedure STP_INSERT_PREVISION_FROM_OS (
  P_FK_EMPRESAS      integer ,
  P_FK_TIPO_ORDENS_SERVICOS smallint,
  P_PK_ORDENS_SERVICOS integer,
  P_DTA_LAN          date,
  P_VLR_LAN          numeric (11, 4),
  P_NUM_PARC         smallint,
  P_TOT_PARC         smallint,
  P_QTD_PARC         smallint
)
returns (
  R_STATUS smallint
)

```



```

as
declare variable FkTipoContas      smallint;
declare variable FkContas          smallint;
declare variable FkCadastrros      integer;
declare variable FkClassificacao   integer;
declare variable FlagTDoc          smallint;
declare variable DscOS             varchar(50);
declare variable DtaLan            date;
declare variable PkContasLancamentos integer;
declare variable PkTipoDocumentos  smallint;
declare variable TypeLan           smallint;
declare variable FlagES            smallint;
declare variable DscLan            varchar(50);

begin
R_STATUS = -1;
TypeLan = 0;
if ((P_FK_EMPRESAS is not null) and (P_FK_EMPRESAS > 0) and
 ( P_FK_TIPO_ORDENS_SERVICOS is not null) and (P_FK_TIPO_ORDENS_SERVICOS > 0) and
 ( P_DTA_LAN is not null) and
 ( P_VLR_LAN is not null) and (P_VLR_LAN > 0) and
 ( P_PK_ORDENS_SERVICOS is not null) and (P_PK_ORDENS_SERVICOS > 0)) then
begin
select Tdc.PK_TIPO_DOCUMENTOS, Tdc.FLAG_TDOC, Grm.FLAG_ES
from TIPO_ORDENS_SERVICOS Tos, TIPO_DOCUMENTOS Tdc,
GRUPOS_MOVIMENTOS Grm
where Tos.PK_TIPO_ORDENS_SERVICOS = :P_FK_TIPO_ORDENS_SERVICOS
and Tdc.PK_TIPO_DOCUMENTOS = Tos.FK_TIPO_DOCUMENTOS
and Grm.PK_GRUPOS_MOVIMENTOS = Tos.FK_GRUPOS_MOVIMENTOS
into :PkTipoDocumentos, :FlagTDoc, :FlagES;
select FK_TIPO_CONTAS, FK_CONTAS
from PARAMETROS
where FK_EMPRESAS = :P_FK_EMPRESAS
into :FkTipoContas, :FkContas;
if ((FkTipoContas is null) or (FkTipoContas = 0) or
 ( FkContas is null) or (FkContas = 0)) then
exit;
select FK_CADASTROS, Cast(DSC_ORD as varchar(50))
from ORDENS_SERVICOS
where FK_EMPRESAS = :P_FK_EMPRESAS
and PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
into :FkCadastrros, :DscLan;
if ((FkCadastrros is null) or (FkCadastrros = 0) or
 ( DscLan is null)) then
exit;
DscOS = Cast(DscLan as varchar(33));
select R_PK_CONTAS_LANCAMENTOS from STP_GET_KC_CONTAS_LANCAMENTOS (
:P_FK_EMPRESAS)
into PkContasLancamentos;
if ((PkContasLancamentos is null) or (PkContasLancamentos = 0)) then
exit;
DscOS = DscOS || ' Parcela: ' || Cast(P_NUM_PARC as varchar(3)) || '/' ||
Cast(P_TOT_PARC as varchar(3));
insert into CONTAS_LANCAMENTOS
(FK_EMPRESAS, PK_CONTAS_LANCAMENTOS, FK_EMPRESAS__CTA, FK_TIPO_CONTAS,
FK_CONTAS, FK_FINALIZADORAS, FK_OPERACOES_FINANCEIRAS, FK_CADASTROS,
FK_HITORICOS_PADRAO, DSC_LAN, DTA_LAN, VLR_LAN, FLAG_DBCR)
values
(:P_FK_EMPRESAS, :PkContasLancamentos, :P_FK_EMPRESAS, :FkTipoContas,

```

```
:FkContas, null, null, :FkCadastrros, null, :DscOS, :P_DTA_LAN,
:P_VLR_LAN, :FLagES);
/* insert or update data to generic document table */
if (FlagTDoc is null) then exit;
TypeLan = null;
select FLAG_TDOC
from VINCULO_DOCUMENTOS
where FK_EMPRESAS      = :P_FK_EMPRESAS
and FK_DOCUMENTOS     = :P_PK_ORDENS_SERVICOS
and FK_CADASTROS      = :FkCadastrros
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_ORDENS_SERVICOS
into :TypeLan;
if (TypeLan is null) then
begin
  TypeLan = 0;
  insert into VINCULO_DOCUMENTOS
    (FK_EMPRESAS, FK_DOCUMENTOS, FK_CADASTROS, FK_TIPO_DOCUMENTOS, FLAG_TDOC)
  values
    (:P_FK_EMPRESAS, :P_PK_ORDENS_SERVICOS, :FkCadastrros,
    :P_FK_TIPO_ORDENS_SERVICOS, :FlagTDoc);
end
/* insert a link between generic document and flow cash */
insert into CONTAS_DOCUMENTOS
(FK_EMPRESAS, FK_CONTAS_LANCAMENTOS, FK_DOCUMENTOS, FK_TIPO_DOCUMENTOS,
FK_CADASTROS, FLAG_TDOC)
values
(:P_FK_EMPRESAS, :PkContasLancamentos, :P_PK_ORDENS_SERVICOS,
:P_FK_TIPO_ORDENS_SERVICOS, :FkCadastrros, :FlagTDoc);
/* insert links between generic document and finance classification */
for select Osi.FK_CLASSIFICACAO, Osi.TOT_SRV, Os.DTA_OS, Os.FLAG_LPREV
from ORDENS_SERVICOS Os, ORDENS_SERVICOS_ITENS Osi
where Os.FK_EMPRESAS      = :P_FK_EMPRESAS
and Os.PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
and Osi.FK_EMPRESAS      = Os.FK_EMPRESAS
and Osi.FK_ORDENS_SERVICOS = Os.PK_ORDENS_SERVICOS
into :FkClassificacao, :P_VLR_LAN, :DtaLan, :TypeLan do
begin
  if ((P_QTD_PARC <= 0) and (TypeLan > 0)) then exit;
  if ((FkClassificacao is not null) and (FkClassificacao > 0)) then
  begin
    if (P_QTD_PARC > 0) then
      P_VLR_LAN = P_VLR_LAN / P_QTD_PARC;
    else
      P_DTA_LAN = DtaLan;
    if (P_QTD_PARC > 0) then
      DscLan = DscOS;
    insert into CLASSIFICACAO_DOCUMENTOS
      (FK_EMPRESAS, FK_CLASSIFICACAO, FK_DOCUMENTOS, FK_TIPO_DOCUMENTOS,
      FK_CADASTROS, FLAG_TDOC, FLAG_PREV, DTA_LAN, VLR_LAN, DSC_LAN,
      PK_CLASSIFICACAO_DOCUMENTOS, FLAG_DBCR)
    values
      (:P_FK_EMPRESAS, :FkClassificacao, :P_PK_ORDENS_SERVICOS,
      :P_FK_TIPO_ORDENS_SERVICOS, :FkCadastrros, :FlagTDoc, 1, :P_DTA_LAN,
      :P_VLR_LAN, :DscLan, 0, :FLagES);
    if ((P_QTD_PARC <= 0) and (TypeLan = 0)) then
    begin
      update ORDENS_SERVICOS set
        FLAG_LPREV = 1
```

```

where FK_EMPRESAS      = :P_FK_EMPRESAS
and FK_TIPO_ORDENS_SERVICOS = :P_FK_TIPO_ORDENS_SERVICOS
and PK_ORDENS_SERVICOS   = :P_PK_ORDENS_SERVICOS;
end
end
end
R_STATUS = 0;
end
suspend;
end ^

set term ^;

```

grant EXECUTE on procedure STP_INSERT_PREVISION_FROM_OS to PUBLIC;

1.4.10.3 Text of procedure STP_INSERT_PREVISION_FROM_OS

```

set term ^;

create procedure %PROC% (
  P_FK_EMPRESAS      integer ,
  P_FK_TIPO_ORDENS_SERVICOS smallint,
  P_PK_ORDENS_SERVICOS integer,
  P_DTA_LAN          date,
  P_VLR_LAN          numeric (11, 4),
  P_NUM_PARC         smallint,
  P_TOT_PARC         smallint,
  P_QTD_PARC         smallint
)
returns (
  R_STATUS smallint
)
as
declare variable FkTipoContas      smallint;
declare variable FkContas          smallint;
declare variable FkCadastrros      integer;
declare variable FkClassificacao   integer;
declare variable FlagTDoc          smallint;
declare variable DscOS             varchar(50);
declare variable DtaLan            date;
declare variable PkContasLancamentos integer;
declare variable PkTipoDocumentos  smallint;
declare variable TypeLan           smallint;
declare variable FlagES            smallint;
declare variable DscLan            varchar(50);
begin
  R_STATUS = -1;
  TypeLan = 0;
  if ((P_FK_EMPRESAS      is not null) and (P_FK_EMPRESAS      > 0) and
    ( P_FK_TIPO_ORDENS_SERVICOS is not null) and (P_FK_TIPO_ORDENS_SERVICOS > 0) and
    ( P_DTA_LAN          is not null) and
    ( P_VLR_LAN          is not null) and (P_VLR_LAN          > 0) and
    ( P_PK_ORDENS_SERVICOS   is not null) and (P_PK_ORDENS_SERVICOS   > 0)) then
  begin
    select Tdc.PK_TIPO_DOCUMENTOS, Tdc.FLAG_TDOC, Grm.FLAG_ES
    from TIPO_ORDENS_SERVICOS Tos, TIPO_DOCUMENTOS Tdc,
         GRUPOS_MOVIMENTOS Grm
    where Tos.PK_TIPO_ORDENS_SERVICOS = :P_FK_TIPO_ORDENS_SERVICOS

```

```
and Tdc.PK_TIPO_DOCUMENTOS = Tos.FK_TIPO_DOCUMENTOS
and Grm.PK_GRUPOS_MOVIMENTOS = Tos.FK_GRUPOS_MOVIMENTOS
into :PkTipoDocumentos, :FlagTDoc, :FlagES;
select FK_TIPO_CONTAS, FK_CONTAS
from PARAMETROS
where FK_EMPRESAS = :P_FK_EMPRESAS
into :FkTipoContas, :FkContas;
if ((FkTipoContas is null) or (FkTipoContas = 0) or
    (FkContas is null) or (FkContas = 0)) then
    exit;
select FK_CADASTROS, Cast(DSC_ORD as varchar(50))
from ORDENS_SERVICOS
where FK_EMPRESAS = :P_FK_EMPRESAS
and PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
into :FkCadastrs, :DscLan;
if ((FkCadastrs is null) or (FkCadastrs = 0) or
    (DscLan is null)) then
    exit;
DscOS = Cast(DscLan as varchar(33));
select R_PK_CONTAS_LANCAMENTOS from STP_GET_KC_CONTAS_LANCAMENTOS (
    :P_FK_EMPRESAS)
into PkContasLancamentos;
if ((PkContasLancamentos is null) or (PkContasLancamentos = 0)) then
    exit;
DscOS = DscOS || ' Parcela: ' || Cast(P_NUM_PARC as varchar(3)) || ' ' ||
    Cast(P_TOT_PARC as varchar(3));
insert into CONTAS_LANCAMENTOS
(FK_EMPRESAS, PK_CONTAS_LANCAMENTOS, FK_EMPRESAS_CTA, FK_TIPO_CONTAS,
FK_CONTAS, FK_FINALIZADORAS, FK_OPERACOES_FINANCEIRAS, FK_CADASTROS,
FK_HITORICOS_PADRAO, DSC_LAN, DTA_LAN, VLR_LAN, FLAG_DBCR)
values
(:P_FK_EMPRESAS, :PkContasLancamentos, :P_FK_EMPRESAS, :FkTipoContas,
:FkContas, null, null, :FkCadastrs, null, :DscOS, :P_DTA_LAN,
:P_VLR_LAN, :FLagES);
/* insert or update data to generic document table */
if (FlagTDoc is null) then exit;
TypeLan = null;
select FLAG_TDOC
from VINCULO_DOCUMENTOS
where FK_EMPRESAS = :P_FK_EMPRESAS
and FK_DOCUMENTOS = :P_PK_ORDENS_SERVICOS
and FK_CADASTROS = :FkCadastrs
and FK_TIPO_DOCUMENTOS = :P_FK_TIPO_ORDENS_SERVICOS
into :TypeLan;
if (TypeLan is null) then
begin
    TypeLan = 0;
    insert into VINCULO_DOCUMENTOS
    (FK_EMPRESAS, FK_DOCUMENTOS, FK_CADASTROS, FK_TIPO_DOCUMENTOS, FLAG_TDOC)
    values
    (:P_FK_EMPRESAS, :P_PK_ORDENS_SERVICOS, :FkCadastrs,
    :P_FK_TIPO_ORDENS_SERVICOS, :FlagTDoc);
end
/* insert a link between generic document and flow cash */
insert into CONTAS_DOCUMENTOS
(FK_EMPRESAS, FK_CONTAS_LANCAMENTOS, FK_DOCUMENTOS, FK_TIPO_DOCUMENTOS,
FK_CADASTROS, FLAG_TDOC)
values
```

```

(:P_FK_EMPRESAS, :PkContasLancamentos, :P_PK_ORDENS_SERVICOS,
:P_FK_TIPO_ORDENS_SERVICOS, :FkCadastrros, :FlagTDoc);
/* insert links between generic document and finance classification */
for select Osi.FK_CLASSIFICACAO, Osi.TOT_SRV, Os.DTA_OS, Os.FLAG_LPREV
  from ORDENS_SERVICOS Os, ORDENS_SERVICOS_ITENS Osi
  where Os.FK_EMPRESAS      = :P_FK_EMPRESAS
     and Os.PK_ORDENS_SERVICOS = :P_PK_ORDENS_SERVICOS
     and Osi.FK_EMPRESAS      = Os.FK_EMPRESAS
     and Osi.FK_ORDENS_SERVICOS = Os.PK_ORDENS_SERVICOS
  into :FkClassificacao, :P_VLR_LAN, :DtaLan, :TypeLan do
begin
  if ((P_QTD_PARC <= 0) and (TypeLan > 0)) then exit;
  if ((FkClassificacao is not null) and (FkClassificacao > 0)) then
  begin
    if (P_QTD_PARC > 0) then
      P_VLR_LAN = P_VLR_LAN / P_QTD_PARC;
    else
      P_DTA_LAN = DtaLan;
    if (P_QTD_PARC > 0) then
      DscLan = DscOS;
    insert into CLASSIFICACAO_DOCUMENTOS
      (FK_EMPRESAS, FK_CLASSIFICACAO, FK_DOCUMENTOS, FK_TIPO_DOCUMENTOS,
      FK_CADASTROS, FLAG_TDOC, FLAG_PREV, DTA_LAN, VLR_LAN, DSC_LAN,
      PK_CLASSIFICACAO_DOCUMENTOS, FLAG_DBCR)
    values
      (:P_FK_EMPRESAS, :FkClassificacao, :P_PK_ORDENS_SERVICOS,
      :P_FK_TIPO_ORDENS_SERVICOS, :FkCadastrros, :FlagTDoc, 1, :P_DTA_LAN,
      :P_VLR_LAN, :DscLan, 0, :FLagES);
    if ((P_QTD_PARC <= 0) and (TypeLan = 0)) then
    begin
      update ORDENS_SERVICOS set
        FLAG_LPREV = 1
      where FK_EMPRESAS      = :P_FK_EMPRESAS
         and FK_TIPO_ORDENS_SERVICOS = :P_FK_TIPO_ORDENS_SERVICOS
         and PK_ORDENS_SERVICOS      = :P_PK_ORDENS_SERVICOS;
    end
  end
end
R_STATUS = 0;
end
suspend;
end ^

set term ;^

```

1.4.10.4 List of diagrams containing the procedure STP_INSERT_PREVISION_FROM_OS

Name	Code
SysServ	SYSSERV

1.4.10.5 List of permissions of the procedure STP_INSERT_PREVISION_FROM_OS

User	Grant	Grant With Option	Revoke	Revoke With Cascade
PUBLIC	EXECUTE			