

Report

Schleder
04/04/2006

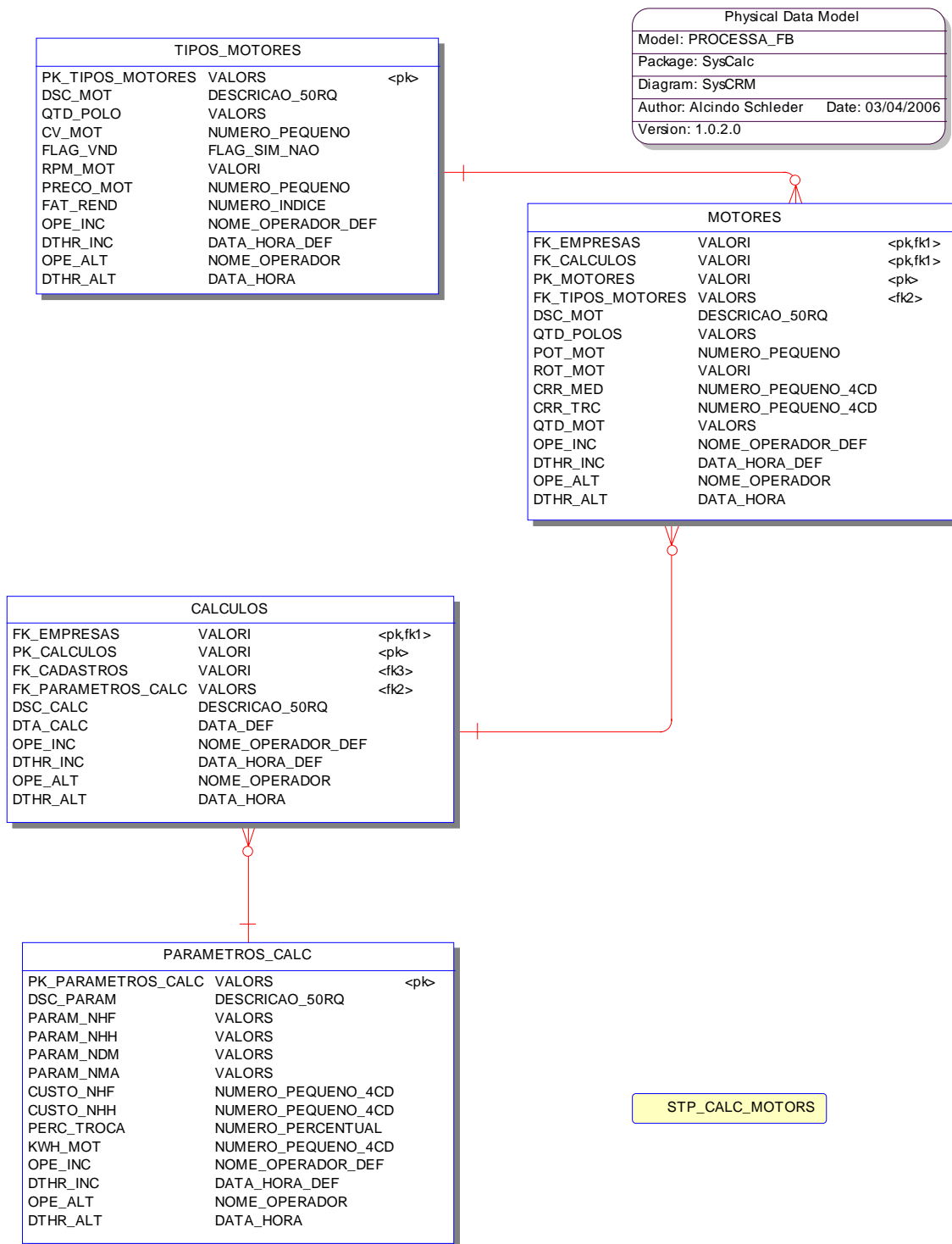
Conteúdo

1 Pacote SysCalc.....	3
1.1 Diagrama SysCRM.....	3
1.1.1 Card of diagram SysCRM.....	5
1.2 List of tables.....	5
1.3 Tabelas.....	6
1.3.1 Tabela CALCULOS.....	6
1.3.2 Tabela MOTORES.....	8
1.3.3 Tabela PARAMETROS_CALC.....	10
1.3.4 Tabela TIPOS_MOTORES.....	12
1.4 Stored Procedures.....	14
1.4.1 Procedure STP_CALC_MOTORS.....	14

1 Pacote SysCalc

1.1 Diagrama SysCRM

Módulo de Gerenciamento de Cálculo



1.1.1 Card of diagram SysCRM

<i>Name</i>	SysCRM
<i>Code</i>	SYSCRM
<i>Comment</i>	

1.2 List of tables

Name	Comment
TIPOS_MOTORES	Tabela que armazena os tipos de motores a serem comparados
PARAMETROS_CALC	Tabela que armazena os parâmetros para os cálculos do consumo de energia
CALCULOS	Tabela que armazena os motores os cadastros e os parâmetros de cálculo para que possa gerar o relatório de cálculo
MOTORES	Tabela que armazena os motores que irão fazer parte do cálculo da planilha.

1.3 Tabelas

1.3.1 Tabela CALCULOS

1.3.1.1 Card of table CALCULOS

Name	CALCULOS
Code	CALCULOS
Comment	Tabela que armazena os motores os cadastros e os parâmetros de cálculo para que possa gerar o relatório de cálculo
Primary Key	PK_CALCULOS

1.3.1.2 Lista de Colunas da Tabela CALCULOS

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
PK_CALCULOS	INTEGER	VALORI	Código da Planilha
FK_CADASTROS	INTEGER	VALORI	Código do Cadastro
FK_PARAMETROS_CALC	SMALLINT	VALORS	Código
DSC_CALC	VARCHAR(50)	DESCRICAO_50RQ	Descrição do cálculo
DTA_CALC	DATE	DATA_DEF	Data em que foi gerado o orçamento
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 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 CALCULOS

Name	Code	Class Name
FK_MOTORES_CALCULOS	FK_MOTORES_CALCULOS	Reference
FK_CALCULOS_EMPRESAS	FK_CALCULOS_EMPRESAS	Reference
FK_CALCULOS_PARAMETROS_CA	FK_CALCULOS_PARAMETROS_CA	Reference
FK_CALCULOS_CADASTROS	FK_CALCULOS_CADASTROS	Reference

1.3.1.4 Lista de Chaves da Tabela CALCULOS

Name	Code	Primary
PK_CALCULOS	PK_CALCULOS	X

1.3.1.5 Chave PK_CALCULOS da Tabela CALCULOS

1.3.1.5.1 Lista de Colunas da Chave PK_CALCULOS

Name	Code
FK_EMPRESAS	FK_EMPRESAS
PK_CALCULOS	PK_CALCULOS

1.3.1.5.2 Dependências da Chave PK_CALCULOS

Name	Code	Class Name
PK_CALCULOS	PK_CALCULOS	Index
FK_MOTORES_CALCULOS	FK_MOTORES_CALCULOS	Reference

1.3.1.6 Lista de Índices CALCULOS

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

1.3.1.7 Índice PK_CALCULOS da Tabela CALCULOS

1.3.1.7.1 Lista de Colunas do Índice PK_CALCULOS

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
PK_CALCULOS	PK_CALCULOS	Ascending

1.3.1.8 Lista de Triggers da Tabela CALCULOS

Name	Code
TBU0_CALCULOS	TBU0_CALCULOS
TBIG_CALCULOS	TBIG_CALCULOS

1.3.2 Tabela MOTORES

1.3.2.1 Card of table MOTORES

Name	MOTORES
Code	MOTORES
Comment	Tabela que armazena os motores que irão fazer parte do cálculo da planilha.
Primary Key	PK_MOTORES

1.3.2.2 Lista de Colunas da Tabela MOTORES

Name	Data Type	Domain	Comment
FK_EMPRESAS	INTEGER	VALORI	Código da Empresa
FK_CALCULOS	INTEGER	VALORI	Código da Planilha
PK_MOTORES	INTEGER	VALORI	Código sequencial da Planilha
FK_TIPOS_MOTORES	SMALLINT	VALORS	
DSC_MOT	VARCHAR(50)	DESCRICAO_50RQ	Armazena uma descrição para o motor
QTD_POLOS	SMALLINT	VALORS	Quantidade de polos do motor do cliente
POT_MOT	NUMERIC(9,2)	NUMERO_PEQUENO	Potência nominal do motor em CV
ROT_MOT	INTEGER	VALORI	Número de rotações do motor do cliente
CRR_MED	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Corrente medida no motor
CRR_TRC	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Corrente teórica informada no manual do motor
QTD_MOT	SMALLINT	VALORS	Quantidade de Motores
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 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 MOTORES

Name	Code	Class Name
FK_MOTORES_CALCULOS	FK_MOTORES_CALCULOS	Reference
FK_MOTORES_TIPOS_MOTORES	FK_MOTORES_TIPOS_MOTORES	Reference

1.3.2.4 Lista de Chaves da Tabela MOTORES

Name	Code	Primary
PK_MOTORES	PK_MOTORES	X

1.3.2.5 Chave PK_MOTORES da Tabela MOTORES**1.3.2.5.1 Lista de Colunas da Chave PK_MOTORES**

Name	Code
FK_EMPRESAS	FK_EMPRESAS
FK_CALCULOS	FK_CALCULOS
PK_MOTORES	PK_MOTORES

1.3.2.5.2 Dependências da Chave PK_MOTORES

Name	Code	Class Name
PK_MOTORES	PK_MOTORES	Index

1.3.2.6 Lista de Índices MOTORES

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

1.3.2.7 Índice PK_MOTORES da Tabela MOTORES**1.3.2.7.1 Lista de Colunas do Índice PK_MOTORES**

Name	Code	Sort
FK_EMPRESAS	FK_EMPRESAS	Ascending
FK_CALCULOS	FK_CALCULOS	Ascending
PK_MOTORES	PK_MOTORES	Ascending

1.3.2.8 Lista de Triggers da Tabela MOTORES

Name	Code
TBU0_MOTORES	TBU0_MOTORES
TBIG_MOTORES	TBIG_MOTORES

1.3.3 Tabela PARAMETROS_CALC

1.3.3.1 Card of table PARAMETROS_CALC

Name	PARAMETROS_CALC
Code	PARAMETROS_CALC
Comment	Tabela que armazena os parâmetros para os cálculos do consumo de energia
Primary Key	PK_PARAMETROS_CALC

1.3.3.2 Lista de Colunas da Tabela PARAMETROS_CALC

Name	Data Type	Domain	Comment
PK_PARAMETROS_CALC	SMALLINT	VALORS	Código
DSC_PARAM	VARCHAR(50)	DESCRICAO_50RQ	Descrição dos parâmetros
PARAM_NHF	SMALLINT	VALORS	Número de horas trabalhadas Fora do horário de Ponta
PARAM_NH	SMALLINT	VALORS	Número de horas trabalhadas dentro do horário de Ponta
PARAM_NDM	SMALLINT	VALORS	Número de dias trabalhados no mês
PARAM_NMA	SMALLINT	VALORS	Número de meses trabalhados no ano
CUSTO_NHF	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Custo da energia no horário de ponta
CUSTO_NH	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Custo da energia nos horários fora de ponta
PERC_TROCA	NUMERIC(5,2)	NUMERO_PERCENTUAL	Percentual de desconto na base de troca
KWH_MOT	NUMERIC(9,4)	NUMERO_PEQUENO_4CD	Kilowats hora do motor
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 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 PARAMETROS_CALC

Name	Code	Class Name
FK_CALCULOS_PARAMETROS_CA	FK_CALCULOS_PARAMETROS_CA	Reference

1.3.3.4 Lista de Chaves da Tabela PARAMETROS_CALC

Name	Code	Primary
PK_PARAMETROS_CALC	PK_PARAMETROS_CALC	X

1.3.3.5 Chave PK_PARAMETROS_CALC da Tabela PARAMETROS_CALC

1.3.3.5.1 Lista de Colunas da Chave PK_PARAMETROS_CALC

Name	Code
PK_PARAMETROS_CALC	PK_PARAMETROS_CALC

1.3.3.5.2 Dependências da Chave PK_PARAMETROS_CALC

Name	Code	Class Name
PK_PARAMETROS_CALC	PK_PARAMETROS_CALC	Index
FK_CALCULOS_PARAMETROS_CA A	FK_CALCULOS_PARAMETROS_C A	Reference

1.3.3.6 Lista de Índices PARAMETROS_CALC

Name	Code	Unique	Cluster	Primary	Foreign Key	Alternat e Key
PK_PARAMET ROS_CALC	PK_PARAMET ROS_CALC	X		X		

1.3.3.7 Índice PK_PARAMETROS_CALC da Tabela PARAMETROS_CALC**1.3.3.7.1 Lista de Colunas do Índice PK_PARAMETROS_CALC**

Name	Code	Sort
PK_PARAMETROS_CALC	PK_PARAMETROS_CALC	Ascending

1.3.3.8 Lista de Triggers da Tabela PARAMETROS_CALC

Name	Code
TBIG_PARAMETROS_CALC	TBIG_PARAMETROS_CALC
TBU0_PARAMETROS_CALC	TBU0_PARAMETROS_CALC

1.3.4 Tabela TIPOS_MOTORES

1.3.4.1 Card of table TIPOS_MOTORES

Name	TIPOS_MOTORES
Code	TIPOS_MOTORES
Comment	Tabela que armazena os tipos de motores a serem comparados
Primary Key	PK_TIPOS_MOTORES

1.3.4.2 Lista de Colunas da Tabela TIPOS_MOTORES

Name	Data Type	Domain	Comment
PK_TIPOS_MOTORES	SMALLINT	VALORS	
DSC_MOT	VARCHAR(50)	DESCRICAO_50RQ	Descrição dos motores
QTD_POLO	SMALLINT	VALORS	Qauntidade de Pólos
CV_MOT	NUMERIC(9,2)	NUMERO_PEQUENO	
FLAG_VND	SMALLINT	FLAG_SIM_NAO	Falg que indica se o motor eh um motor de venda
RPM_MOT	INTEGER	VALORI	Número de rotações por minuto do motor
PRECO_MOT	NUMERIC(9,2)	NUMERO_PEQUENO	Preco do Motor
FAT_REND	NUMERIC(9,4)	NUMERO_INDICE	Fator de rendimento do motor é um índice que é um multiplicador do consumo fora do horário de ponta e dentro do horário de ponta.
OPE_INC	VARCHAR(10)	NOME_OPERADOR_DEF	Nome do oeprador 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 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 TIPOS_MOTORES

Name	Code	Class Name
FK_MOTORES_TIPOS_MOTORES	FK_MOTORES_TIPOS_MOTORES	Reference

1.3.4.4 Lista de Chaves da Tabela TIPOS_MOTORES

Name	Code	Primary
PK_TIPOS_MOTORES	PK_TIPOS_MOTORES	X

1.3.4.5 Chave PK_TIPOS_MOTORES da Tabela TIPOS_MOTORES

1.3.4.5.1 Lista de Colunas da Chave PK_TIPOS_MOTORES

Name	Code
PK_TIPOS_MOTORES	PK_TIPOS_MOTORES

1.3.4.5.2 Dependências da Chave PK_TIPOS_MOTORES

Name	Code	Class Name
PK_TIPOS_MOTORES	PK_TIPOS_MOTORES	Index
FK_MOTORES_TIPOS_MOTORES	FK_MOTORES_TIPOS_MOTORES	Reference

1.3.4.6 Lista de Índices TIPOS_MOTORES

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

1.3.4.7 Índice PK_TIPOS_MOTORES da Tabela TIPOS_MOTORES**1.3.4.7.1 Lista de Colunas do Índice PK_TIPOS_MOTORES**

Name	Code	Sort
PK_TIPOS_MOTORES	PK_TIPOS_MOTORES	Ascending

1.3.4.8 Lista de Triggers da Tabela TIPOS_MOTORES

Name	Code
TBIG_TIPOS_MOTORES	TBIG_TIPOS_MOTORES
TBU0_TIPOS_MOTORES	TBU0_TIPOS_MOTORES

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

<i>Name</i>	STP_CALC_MOTORS
<i>Code</i>	STP_CALC_MOTORS
<i>Parent</i>	Package 'SysCalc'
<i>Type</i>	Procedure

1.4.1.2 Code preview of procedure STP_CALC_MOTORS

set term ^;

```

create procedure STP_CALC_MOTORS (
    P_FK_EMPRESAS    INTEGER,
    P_PK_CALCULOS    INTEGER
)
RETURNS (
    R_NOM_CLI        VARCHAR(50),
    R_FON_CLI        VARCHAR(20),
    R_DSC_MOT        VARCHAR(50),
    R_POT_MOT        NUMERIC(9,4),
    R_ROT_MOT        INTEGER,
    R_LOC_MOT        VARCHAR(30),
    R_DSC_MOT_VND    VARCHAR(50),
    R_CNS_PONTA_ATU  DOUBLE PRECISION,
    R_CNS_FPONTA_ATU DOUBLE PRECISION,
    R_CNS_TOTAL_ATU  DOUBLE PRECISION,
    R_CNS_PONTA_NEW  DOUBLE PRECISION,
    R_CNS_FPONTA_NEW DOUBLE PRECISION,
    R_CNS_TOTAL_NEW  DOUBLE PRECISION,
    R_ECON_ENERG     DOUBLE PRECISION,
    R_ECON_ENERGF    DOUBLE PRECISION,
    R_VALOR_ECON     DOUBLE PRECISION,
    R_VALOR_ECONF    DOUBLE PRECISION,
    R_VALOR_ECONT    DOUBLE PRECISION,
    R_RET_INV_ANO_NEW NUMERIC(18,2),
    R_RET_INV_MES_NEW NUMERIC(18,2),
    R_RET_INV_ANO_ATU NUMERIC(18,2),
    R_RET_INV_MES_ATU NUMERIC(18,2),
    R_STATUS         SMALLINT)

```

AS

```

declare variable FkCadastrs    integer;
declare variable FkParametrosCalc integer;
declare variable FkTipoMotores integer;
declare variable QtdPolo       smallint;
declare variable KwhMot        double precision;
declare variable QtdPoloVnd    smallint;
declare variable CvMotVnd      double precision;
declare variable Nhf           smallint;
declare variable Nhh           smallint;
declare variable Ndm           smallint;
declare variable Nma           smallint;
declare variable CustoSux      double precision;
declare variable CustoSux      double precision;

```

```
declare variable CustoNhh      double precision;
declare variable PrecoMot      double precision;
declare variable PercTroca    double precision;
declare variable CorrMed      double precision;
declare variable CorrTeor     double precision;
declare variable Indice       double precision;
declare variable FatRendNew    double precision;
begin
  R_STATUS = -1;
  /* select tabsheet data */
  select FK_CADASTROS, FK_PARAMETROS_CALC
    from CALCULOS
  where FK_EMPRESAS = :P_FK_EMPRESAS
    and PK_CALCULOS = :P_PK_CALCULOS
    into :FkCadastrros, :FkParametrosCalc;
  /* select customer data */
  select RAZ_SOC, FON_CAD from VW_CLIENTES
  where PK_CADASTROS = :FkCadastrros
    into :R_NOM_CLI, :R_FON_CLI;
  /* select parameter data */
  select PARAM_NHF, PARAM_NHH, PARAM_NDM, PARAM_NMA, CUSTO_NHF,
    CUSTO_NHH, PERC_TROCA, KWH_MOT
    from PARAMETROS_CALC
  where PK_PARAMETROS_CALC = :FkParametrosCalc
    into :Nhf, :Nhh, :Ndm, :Nma, :CustoNhf, :CustoNhh, :PercTroca,
      :KwhMot;
  /* select customer motor data */
  CorrMed = 0;
  CorrTeor = 0;
  for select FK_TIPOS_MOTORES, DSC_MOT, QTD_POLOS, POT_MOT, ROT_MOT,
    CRR_MED, CRR_TRC
    from MOTORES
  where FK_EMPRESAS = :P_FK_EMPRESAS
    and FK_CALCULOS = :P_PK_CALCULOS
    into :FkTipoMotores, :R_DSC_MOT, :QtdPolo, :R_POT_MOT, :R_ROT_MOT,
      :CorrMed, :CorrTeor do
  begin
    R_LOC_MOT = ' ';
    if ((CorrMed is nul) or (CorrMed <= 0)) then
      CorrMed = 0;
    if ((CorrTeor is nul) or (CorrTeor <= 0)) then
      CorrTeor = 0;
    if ((CorrTeor > 0) and (CorrMed > 0)) then
      Indice = CorrMed / CorrTeor;
    else
      Indice = 1;
  /* select customer data */
  select RAZ_SOC, FON_CAD from VW_CLIENTES
  where PK_CADASTROS = :FkCadastrros
    into :R_NOM_CLI, :R_FON_CLI;
  /* select sale motor data */
  select DSC_MOT, QTD_POLO, CV_MOT, PRECO_MOT, FAT_REND
    from TIPOS_MOTORES
  where PK_TIPOS_MOTORES = :FkTipoMotores
    into :R_DSC_MOT_VND, :QtdPoloVnd, :CvMotVnd, :PrecoMot,
      :FatRendNew;
  if ((FatRendNew is null) or (FatRendNew <= 0)) then
    FatRendNew = 1;
```

```

/* Consumo no horario de ponta do motor atual */
R_CNS_PONTA_ATU = KwhMot * R_POT_MOT;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Nhh;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Ndm;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Nma;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Indice;
/* Consumo fora do horario de ponta do motor atual */
R_CNS_FPONTA_ATU = KwhMot * R_POT_MOT;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Nhf;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Ndm;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Nma;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Indice;

R_CNS_TOTAL_ATU = R_CNS_PONTA_ATU + R_CNS_FPONTA_ATU;
/* Consumo no horario de ponta do motor proposto */
R_CNS_PONTA_NEW = KwhMot * CvMotVnd;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Nhh;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Ndm;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Nma;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * FatRendNew;
/* Consumo fora do horario de ponta do motor proposto */
R_CNS_FPONTA_NEW = KwhMot * CvMotVnd;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Nhf;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Ndm;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Nma;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * FatRendNew;

R_CNS_TOTAL_NEW = R_CNS_PONTA_ATU + R_CNS_FPONTA_NEW;
/* Economia de Energia no horario de ponta */
R_ECON_ENERG = R_CNS_PONTA_ATU - R_CNS_PONTA_NEW;
R_VALOR_ECON = R_ECON_ENERG * CustoNhh;
/* Economia de Energia fora do horario de ponta */
R_ECON_ENERGF = R_CNS_FPONTA_ATU - R_CNS_FPONTA_NEW;
R_VALOR_ECONF = R_ECON_ENERGF * CustoNhf;
R_VALOR_ECONT = R_VALOR_ECONF + R_VALOR_ECON;
/* Retorno do investimento com aquisicao e descarte */
R_RET_INV_ANO_NEW = 0;
R_RET_INV_MES_NEW = 0;
if (R_VALOR_ECONT > 0) then
begin
  R_RET_INV_ANO_NEW = PrecoMot / R_VALOR_ECONT;
  R_RET_INV_MES_NEW = R_RET_INV_ANO_NEW * Nma;
end
/* Retorno do investimento a base de troca */
R_RET_INV_ANO_ATU = 0;
R_RET_INV_MES_ATU = 0;
if (R_VALOR_ECONT > 0) then
begin
  CustoAux = (PrecoMot * PercTroca) / 100;
  PrecoMot = PrecoMot - CustoAux;
  R_RET_INV_ANO_ATU = PrecoMot / R_VALOR_ECONT;
  R_RET_INV_MES_ATU = R_RET_INV_ANO_ATU * Nma;
end
R_STATUS = 0;
suspend;
end
end ^

```


set term ^;

grant execute on procedure STP_CALC_MOTORS to public;

1.4.1.3 Text of procedure STP_CALC_MOTORS

set term ^;

```
create procedure %PROC% (  
    P_FK_EMPRESAS    INTEGER,  
    P_PK_CALCULOS    INTEGER  
)  
RETURNS (  
    R_NOM_CLI        VARCHAR(50),  
    R_FON_CLI        VARCHAR(20),  
    R_DSC_MOT        VARCHAR(50),  
    R_POT_MOT        NUMERIC(9,4),  
    R_ROT_MOT        INTEGER,  
    R_LOC_MOT        VARCHAR(30),  
    R_DSC_MOT_VND    VARCHAR(50),  
    R_CNS_PONTA_ATU  DOUBLE PRECISION,  
    R_CNS_FPONTA_ATU DOUBLE PRECISION,  
    R_CNS_TOTAL_ATU  DOUBLE PRECISION,  
    R_CNS_PONTA_NEW  DOUBLE PRECISION,  
    R_CNS_FPONTA_NEW DOUBLE PRECISION,  
    R_CNS_TOTAL_NEW  DOUBLE PRECISION,  
    R_ECON_ENERG     DOUBLE PRECISION,  
    R_ECON_ENERGF    DOUBLE PRECISION,  
    R_VALOR_ECON     DOUBLE PRECISION,  
    R_VALOR_ECONF    DOUBLE PRECISION,  
    R_VALOR_ECONT    DOUBLE PRECISION,  
    R_RET_INV_ANO_NEW NUMERIC(18,2),  
    R_RET_INV_MES_NEW NUMERIC(18,2),  
    R_RET_INV_ANO_ATU NUMERIC(18,2),  
    R_RET_INV_MES_ATU NUMERIC(18,2),  
    R_STATUS         SMALLINT)
```

AS

```
declare variable FkCadastrs    integer;  
declare variable FkParametrosCalc integer;  
declare variable FkTipoMotores integer;  
declare variable QtdPolo       smallint;  
declare variable KwhMot        double precision;  
declare variable QtdPoloVnd    smallint;  
declare variable CvMotVnd      double precision;  
declare variable Nhf           smallint;  
declare variable Nhh           smallint;  
declare variable Ndm           smallint;  
declare variable Nma           smallint;  
declare variable CustoSux      double precision;  
declare variable CustoSuxNhf   double precision;  
declare variable CustoSuxNhh   double precision;  
declare variable PrecoMot      double precision;  
declare variable PercTroca     double precision;  
declare variable CorrMed       double precision;  
declare variable CorrTeor      double precision;  
declare variable Indice        double precision;  
declare variable FatRendNew     double precision;  
begin  
    R_STATUS = -1;
```

```
/* select tabsheet data */
select FK_CADASTROS, FK_PARAMETROS_CALC
  from CALCULOS
 where FK_EMPRESAS = :P_FK_EMPRESAS
   and PK_CALCULOS = :P_PK_CALCULOS
 into :FkCadastrOs, :FkParametrosCalc;
/* select customer data */
select RAZ_SOC, FON_CAD from VW_CLIENTES
 where PK_CADASTROS = :FkCadastrOs
 into :R_NOM_CLI, :R_FON_CLI;
/* select parameter data */
select PARAM_NHF, PARAM_NHH, PARAM_NDM, PARAM_NMA, CUSTO_NHF,
       CUSTO_NHH, PERC_TROCA, KWH_MOT
  from PARAMETROS_CALC
 where PK_PARAMETROS_CALC = :FkParametrosCalc
 into :Nhf, :Nhh, :Ndm, :Nma, :CustoNhf, :CustoNhh, :PercTroca,
       :KwhMot;
/* select customer motor data */
CorrMed = 0;
CorrTeor = 0;
for select FK_TIPOS_MOTORES, DSC_MOT, QTD_POLOS, POT_MOT, ROT_MOT,
          CRR_MED, CRR_TRC
  from MOTORES
 where FK_EMPRESAS = :P_FK_EMPRESAS
   and FK_CALCULOS = :P_PK_CALCULOS
 into :FkTipoMotores, :R_DSC_MOT, :QtdPolo, :R_POT_MOT, :R_ROT_MOT,
       :CorrMed, :CorrTeor do
begin
  R_LOC_MOT = '_';
  if ((CorrMed is nul) or (CorrMed <= 0)) then
    CorrMed = 0;
  if ((CorrTeor is nul) or (CorrTeor <= 0)) then
    CorrTeor = 0;
  if ((CorrTeor > 0) and (CorrMed > 0)) then
    Indice = CorrMed / CorrTeor;
  else
    Indice = 1;
/* select customer data */
select RAZ_SOC, FON_CAD from VW_CLIENTES
 where PK_CADASTROS = :FkCadastrOs
 into :R_NOM_CLI, :R_FON_CLI;
/* select sale motor data */
select DSC_MOT, QTD_POLO, CV_MOT, PRECO_MOT, FAT_REND
  from TIPOS_MOTORES
 where PK_TIPOS_MOTORES = :FkTipoMotores
 into :R_DSC_MOT_VND, :QtdPoloVnd, :CvMotVnd, :PrecoMot,
       :FatRendNew;
if ((FatRendNew is null) or (FatRendNew <= 0)) then
  FatRendNew = 1;
/* Consumo no horario de ponta do motor atual */
R_CNS_PONTA_ATU = KwhMot * R_POT_MOT;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Nhh;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Ndm;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Nma;
R_CNS_PONTA_ATU = R_CNS_PONTA_ATU * Indice;
/* Consumo fora do horario de ponta do motor atual */
R_CNS_FPONTA_ATU = KwhMot * R_POT_MOT;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Nhf;
```

```

R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Ndm;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Nma;
R_CNS_FPONTA_ATU = R_CNS_FPONTA_ATU * Indice;

R_CNS_TOTAL_ATU = R_CNS_PONTA_ATU + R_CNS_FPONTA_ATU;
/* Consumo no horario de ponta do motor proposto */
R_CNS_PONTA_NEW = KwhMot * CvMotVnd;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Nhh;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Ndm;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * Nma;
R_CNS_PONTA_NEW = R_CNS_PONTA_NEW * FatRendNew;
/* Consumo fora do horario de ponta do motor proposto */
R_CNS_FPONTA_NEW = KwhMot * CvMotVnd;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Nhfh;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Ndm;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * Nma;
R_CNS_FPONTA_NEW = R_CNS_FPONTA_NEW * FatRendNew;

R_CNS_TOTAL_NEW = R_CNS_PONTA_ATU + R_CNS_FPONTA_NEW;
/* Economia de Energia no horario de ponta */
R_ECON_ENERG = R_CNS_PONTA_ATU - R_CNS_PONTA_NEW;
R_VALOR_ECON = R_ECON_ENERG * CustoNhh;
/* Economia de Energia fora do horario de ponta */
R_ECON_ENERGF = R_CNS_FPONTA_ATU - R_CNS_FPONTA_NEW;
R_VALOR_ECONF = R_ECON_ENERGF * CustoNhfh;
R_VALOR_ECONT = R_VALOR_ECONF + R_VALOR_ECON;
/* Retorno do investimento com aquisicao e descarte */
R_RET_INV_ANO_NEW = 0;
R_RET_INV_MES_NEW = 0;
if (R_VALOR_ECONT > 0) then
begin
  R_RET_INV_ANO_NEW = PrecoMot / R_VALOR_ECONT;
  R_RET_INV_MES_NEW = R_RET_INV_ANO_NEW * Nma;
end
/* Retorno do investimento a base de troca */
R_RET_INV_ANO_ATU = 0;
R_RET_INV_MES_ATU = 0;
if (R_VALOR_ECONT > 0) then
begin
  CustoAux = (PrecoMot * PercTroca) / 100;
  PrecoMot = PrecoMot - CustoAux;
  R_RET_INV_ANO_ATU = PrecoMot / R_VALOR_ECONT;
  R_RET_INV_MES_ATU = R_RET_INV_ANO_ATU * Nma;
end
R_STATUS = 0;
suspend;
end
end ^

set term ;^

```

1.4.1.4 List of diagrams containing the procedure STP_CALC_MOTORS

Name	Code
SysCRM	SYSCRM