

Criação do Banco de dados Subiter

Criação da tabela EMPRESA:

```
CREATE TABLE empresa (  
    codigo_empresa INTEGER NOT NULL,  
    cnpj_empresa VARCHAR2(14) NOT NULL,  
    nome_empresa VARCHAR2(60) NOT NULL,  
    endereco_empresa VARCHAR2(60) NOT NULL,  
    contato_empresa VARCHAR2(30) NOT NULL);  
  
ALTER TABLE empresa ADD CONSTRAINT empresa_pk PRIMARY KEY ( codigo_empresa );
```

Criação da tabela USUARIOS:

```
CREATE TABLE usuario ( codigo_usuario INTEGER NOT NULL,  
    nome_usuario VARCHAR2(40) NOT NULL,  
    tipo_usuario VARCHAR2(15 CHAR) NOT NULL,  
    senha_usuario VARCHAR2(10),  
    email_usuario VARCHAR2(40),  
    codigo_empresa INTEGER NOT NULL);  
  
ALTER TABLE usuario ADD CONSTRAINT usuario_pk PRIMARY KEY ( codigo_usuario );  
  
ALTER TABLE usuario  
    ADD CONSTRAINT usuario_empresa_fk FOREIGN KEY ( codigo_empresa )  
    REFERENCES empresa ( codigo_empresa );
```

Criação da tabela TIPO_SERVICO:

```
CREATE TABLE tipo_servico (  
    tipo_servico_codigo INTEGER NOT NULL,  
    tipo_servico_nome VARCHAR2(20) NOT NULL,  
    tipo_servico_descricao VARCHAR2(40));  
  
ALTER TABLE tipo_servico ADD CONSTRAINT tipo_servico_pk PRIMARY KEY ( tipo_servico_codigo );
```

Criação da tabela EQUIPAMENTO:

```
CREATE TABLE equipamento (  
    codigo_equipamento INTEGER NOT NULL,  
    nome_equipamento VARCHAR2(40) NOT NULL,  
    descricao_equipamento VARCHAR2(60),  
    fabricante_equipamento VARCHAR2(30));  
  
ALTER TABLE equipamento ADD CONSTRAINT equipamento_pk PRIMARY KEY (  
    codigo_equipamento );
```

Criação da tabela SERVICO:

```
CREATE TABLE servico (  servico_codigo          INTEGER NOT NULL,  
  
    servico_tipo_servico_codigo INTEGER NOT NULL,  
  
    servico_descricao      VARCHAR2(120) NOT NULL,  
  
    servico_data_inclusao   DATE NOT NULL,  
  
    servico_data_agendamento DATE NOT NULL,  
  
    servico_data_realizacao DATE,  
  
    empresa_codigo_empresa INTEGER NOT NULL);
```

```
ALTER TABLE servico ADD CONSTRAINT servico_pk PRIMARY KEY ( servico_codigo );
```

```
ALTER TABLE servico
```

```
    ADD CONSTRAINT servico_empresa_fk FOREIGN KEY ( empresa_codigo_empresa )  
  
        REFERENCES empresa ( codigo_empresa );
```

```
ALTER TABLE servico
```

```
    ADD CONSTRAINT servico_tipo_servico_fk FOREIGN KEY ( servico_tipo_servico_codigo )  
  
        REFERENCES tipo_servico ( tipo_servico_codigo );
```

Criação da tabela EMPRESA_EQUIPAMENTO:

CREATE TABLE empresa_equipamento (

equipamento_empresa_codigo INTEGER NOT NULL,

equipamento_empresa_serie VARCHAR2(30) NOT NULL,

equipamento_servico_codigo INTEGER NOT NULL);

ALTER TABLE empresa_equipamento

ADD CONSTRAINT empresa_equipamento_pk PRIMARY KEY (equipamento_empresa_codigo,

equipamento_empresa_serie,

equipamento_servico_codigo);

ALTER TABLE empresa_equipamento

ADD CONSTRAINT empresa_equipamento_empresa_fk FOREIGN KEY (
equipamento_empresa_codigo)

REFERENCES empresa (codigo_empresa);

ALTER TABLE empresa_equipamento

ADD CONSTRAINT empresa_equipamento_equipamento_serie_fk FOREIGN KEY (
equipamento_empresa_serie)

REFERENCES equipamento_serie (serie_equipamento);

ALTER TABLE empresa_equipamento

ADD CONSTRAINT empresa_equipamento_servico_fk FOREIGN KEY (
equipamento_servico_codigo) REFERENCES servico (servico_codigo);

Criação da tabela EQUIPAMENTO_SERIE:

```
CREATE TABLE equipamento_serie (  
    serie_equipamento    VARCHAR2(30) NOT NULL,  
    serie_codigo_equipamento INTEGER NOT NULL,  
    serie_disponivel      NUMBER NOT NULL,  
    serie_data_entrada    DATE NOT NULL);  
  
ALTER TABLE equipamento_serie ADD CONSTRAINT equipamento_serie_pk PRIMARY KEY (  
    serie_equipamento );  
  
ALTER TABLE equipamento_serie  
  
    ADD CONSTRAINT equipamento_serie_equipamento_fk FOREIGN KEY (  
        serie_codigo_equipamento )  
  
        REFERENCES equipamento ( codigo_equipamento );
```

Criação da tabela CHAMADO:

```
CREATE TABLE chamado (  
    numero_chamado    INTEGER NOT NULL,  
    usuario_chamado   INTEGER NOT NULL,  
    tipo_chamado_servico INTEGER NOT NULL,  
    criticidade_chamado VARCHAR2(1 CHAR) NOT NULL,  
    data_chamado      DATE NOT NULL,  
    assunto_chamado   VARCHAR2(120 CHAR) NOT NULL,  
    descricao_chamado VARCHAR2(300 CHAR) NOT NULL,  
    situacao_chamado  CHAR(1 CHAR) NOT NULL,  
    solucao_chamado   VARCHAR2(300 CHAR),  
    encerramento_chamado DATE,  
    atendente_chamado INTEGER  
);
```

ALTER TABLE chamado

ADD CHECK (criticidade_chamado IN ('A', 'B', 'M'));

ALTER TABLE chamado

ADD CHECK (situacao_chamado IN ('A', 'F', 'H'));

COMMENT ON COLUMN chamado.numero_chamado IS

'Numero do chamado é chave principal, obrigatorio unico.

';

COMMENT ON COLUMN chamado.usuario_chamado IS

'Usuario que abriu o chamado deve ser obrigatorioamente do tipo cliente e vem do cdaastro de usuarios.'

COMMENT ON COLUMN chamado.tipo_chamado_servico IS

'Conforme acertado vira do tipo de serviço;

Tipo de chamado= Instalação - Vira um prestação de Serviço para uma nova instalação

Tipo de chamado =Manutenção - Vira uma prestação de serviço para um manutenção do sistema e uipamento por exempolo.

Tipo de chamado =Suporte - Pode ser somente uma duvida/orientação sem necessidade de virar uma prestação de serviço.

';

COMMENT ON COLUMN chamado.criticidade_chamado IS

'B=BAIXA

M=MEDIA

A=ALTA

';

COMMENT ON COLUMN chamado.data_chamado IS

'Data em que o chamado foi aberto.

';

COMMENT ON COLUMN chamado.assunto_chamado IS

'Assunto simplificado motivo do chamado.'

COMMENT ON COLUMN chamado.descricao_chamado IS

'Descrição detalhada da solicitação do chamado, seja ele de suporte,manutenção ou instalação';

COMMENT ON COLUMN chamado.situacao_chamado IS

'A = Aberto, ninguém viu o chamado

H =Liberado para cliente/em homologacao

F = Finalizado

';

COMMENT ON COLUMN chamado.solucao_chamado IS

'Solução dada pelo suporte ao chamado , ou se foi gerado um agendamento para atendimento quando o chamado é de Manutenção ou Instalação.

';

COMMENT ON COLUMN chamado.encerramento_chamado IS

'Data em que o chamado foi encerrado com a solução do problema.

';

COMMENT ON COLUMN chamado.atendente_chamado IS

'Atendente do chamadoi,pode ser vazio, porem se preenchido deve ser um usuario da empresa e nunca do cliente.'

```
ALTER TABLE chamado ADD CONSTRAINT chamado_pk PRIMARY KEY ( numero_chamado );
```

```
ALTER TABLE chamado
```

```
ADD CONSTRAINT chamado_tipo_servico_fk FOREIGN KEY ( tipo_chamado_servico )  
REFERENCES tipo_servico ( tipo_servico_codigo );
```

```
ALTER TABLE chamado
```

```
ADD CONSTRAINT chamado_usuario_fk FOREIGN KEY ( usuario_chamado )  
REFERENCES usuario ( codigo_usuario );
```

```
ALTER TABLE chamado
```

```
ADD CONSTRAINT chamado_usuario_fkv1 FOREIGN KEY ( atendente_chamado )  
REFERENCES usuario ( codigo_usuario );
```

Criação da tabela **AGENDAMENTO_CHAMADO**:

```
CREATE TABLE agendamento_chamado (  
    numero_agendamento          INTEGER NOT NULL,  
    numero_chamado_agendamento  INTEGER NOT NULL,  
    datahora_atendimento_agendamento  TIMESTAMP WITH LOCAL TIME ZONE,  
    pessoas_atendimento_agendamento  VARCHAR2(100),  
    descricao_atendimento_agendamento VARCHAR2(300),  
    local_atendimento_agendamento   VARCHAR2(300)  
);
```

```
COMMENT ON COLUMN agendamento_chamado.numero_agendamento IS  
'Numero do agendamento Campo Chave..';
```


COMMENT ON COLUMN agendamento_chamado.numero_chamado_agendamento IS

'Numero do chamado que sera agendado';

COMMENT ON COLUMN agendamento_chamado.datahora_atendimento_agendamento IS

'Data e hora que será feito atendimento';

COMMENT ON COLUMN agendamento_chamado.pessoas_atendimento_agendamento IS

'Colocar livermente nome das pessoas que faraa o atendimento';

COMMENT ON COLUMN agendamento_chamado.descricao_atendimento_agendamento IS

'Descrição dos que srá o atendimento como equipamentos -livre digitação';

COMMENT ON COLUMN agendamento_chamado.local_atendimento_agendamento IS

'Local de atendimento - livre digitação';

**ALTER TABLE agendamento_chamado ADD CONSTRAINT agendamento_chamado_pk PRIMARY KEY
(numero_agendamento);**

ALTER TABLE agendamento_chamado

**ADD CONSTRAINT agendamento_chamado_chamado_fk FOREIGN KEY (
numero_chamado_agendamento)**

REFERENCES chamado (numero_chamado);