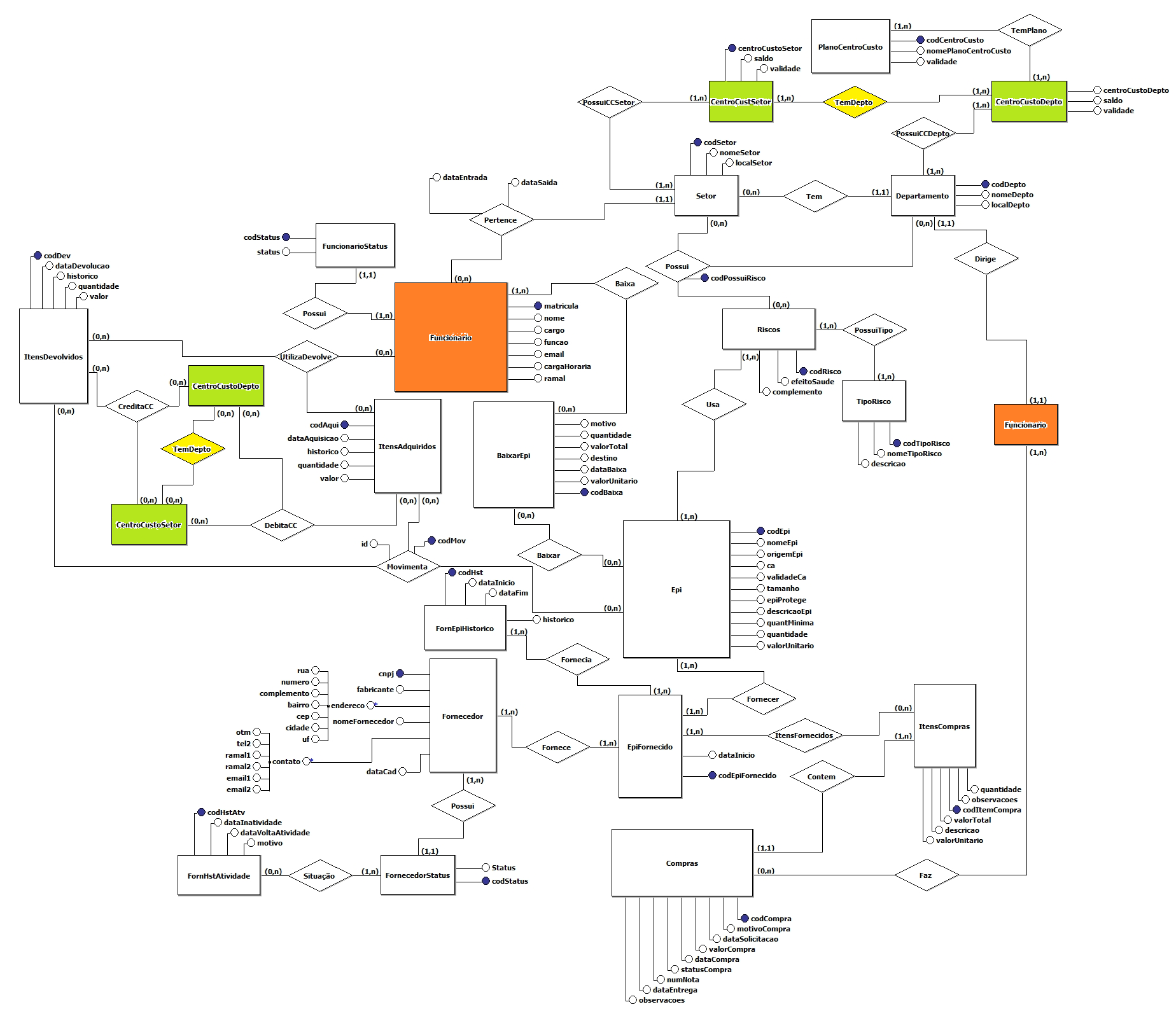
**Projeto de Banco de Dados**

**Diagrama Entidade Relacionamento**

****

**Projeto Lógico**

Departamento (codDepto , nomeDepto, localDepto, matricula)

matricula referencia Funcionário

Setor (codSetor, nomeSetor, localSetor, matricula, codDepto)

CodDepto referencia Departamento

matricula referencia Funcionário

Funcionario (matricula, nome, cargo, funcao, ramal, email, cargaHoraria, codSetor, codDepto, dataEntrada, dataSaida)

codSetor referencia Setor

codDepto referencia Departamento

FuncionarioStatus (codStatus, status, matricula)

matricula referencia Funcionario

EPI (codEpi, nomeEpi, origemEpi, CA, validadeCA, tamanho, epiProtege, descricaoEpi, quantMinima, quantidade, valorUnitario)

TipoRisco (codTipoRisco, nomeTipoRisco, descricao)

Riscos(codRisco, codTipoRisco, efeitoSaude, codEpi, complemento)

codEpi referencia EPI

codTipoRisco referencia TipoRisco

Possui (codPossuiRisco, codRisco, codSetor, codDepto)

codRisco referencia Riscos

codSetor referencia Setor

codDepto referencia Departamento

PlanoCentroCusto (codCentroCusto, nomePlanoCentroCusto, validade)

CentroCustoDepto (centroCustoDepto, codCentroCusto, codDepto, saldo, validade)

codCentroCusto referencia PlanoCentroCusto

codDepto referencia Departamento

CentroCustoSetor (centroCustoSetor, centroCustoDepto, codSetor, saldo, validade)

centroCustoDepto referencia CentroCustoDepto

codSetor referencia Setor

ItensAdquiridos (codAqui, dataAquisicao, historico, quantidade, valor, centroCustoSetor, centroCustoDepto, codEpi, matricula)

centroCustoSetor referencia centroCustoSetor

centroCustoDepto referencia centroCustoDepto

codEpi referencia Epi

matricula referencia Funcionario

ItensDevolvidos (codDev, dataDevolucao, historico, quantidade, valor, centroCustoSetor, centroCustoDepto, codEpi, matricula)

centroCustoSetor referencia centroCustoSetor

centroCustoDepto referencia centroCustoDepto

codEpi referencia Epi

matricula referencia Funcionario

Movimenta (codMov, codAqui, codDev, id)

codAqui referencia ItensAdquiridos

codDev referencia ItensDevolvidos

Fornecedor (cnpj, nomeFornecedor, fabricante, endereco, contato, dataCad)

FornecedorStatus (codStatus, Status, cnpj)

cnpj referencia Fornecedor

FornHstAtividade (codHstAtv, codStatus, dataInatividade, dataVoltaAtividade, motivo)

codStatus referencia FornecedorStatus

EpiFornecido (codEpiFornecido, dataInicio, codEpi, cnpj)

Cnpj referencia Fornecedor

codEpi referencia EPI

FornEpiHistorico (codHst, codEpiFornecido, dataInicio, dataFim, historico)

codEpiFornecido referencia EpiFornecido

Compras (codCompra, matricula, dataSolicitacao, dataCompra, motivoCompra, valorCompra, statusCompra, numNota, dataEntrega, observacoes)

matricula referencia Funcionario

ItemCompras (codItemCompra, codCompra, codEpiFornecido, quantidade, valorUnitario, valorTotal, observacoes)

codCompra referencia Compras

codEpiFornecido referencia EpiFornecido

BaixarEpi (codBaixa, codEpi, matricula, nome, codSetor, motivo, observacoes, quantidade, valorUnitatio, valorTotal, destino, dataBaixa)

codEpi referencia EPI

matricula referencia Funcionario

**Dicionário de Dados**

O dicionário de dados tem como objetivo especificar os metadados que contem as especificações dos elementos de cada tabela do banco de dados.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: DEPARTAMENTO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS DEPARTAMENTOS | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codDepto | Number(3) | N | Código do departamento | S | N | N | N |
| nomeDepto | Varchar2(60) | N | Nome do departamento | N | N | N | N |
| localDepto | Varchar2(60) | N | Local do departamento | N | N | N | N |
| matricula | Number(3) | N | Matricula do Diretor do departamento | N | S | S | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: SETOR  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS SETORES | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codSetor | Number(3) | N | Código do setor | S | N | N | N |
| nomeSetor | Varchar2(60) | N | Nome do setor | N | N | N | N |
| localSetor | Varchar2(60) | N | Localização do setor | N | N | N | N |
| matricula | Number(6) | N | Matricula do Gerente do setor | N | S | N | N |
| codDepto | Number(3) | N | Código do departamento no qual o setor está associado | N | S | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FUNCIONÁRIO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS FUNCIONÁRIOS | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| matricula | Number(6) | N | Matricula do funcionário | S | N | N | N |
| nome | Varchar2(60) | N | Nome do funcionário | N | N | N | N |
| cargo | Varchar2(50) | N | Cargo do funcionário | N | N | N | N |
| funcao | Varchar2(50) | N | Função do funcionário | N | N | N | N |
| ramal | Number(6) | Y | Ramal do setor ou departamento onde o funcionário trabalha | N | N | N | N |
| email | Varchar2(30) | Y | E-mail do funcionário | N | N | N | N |
| cargaHoraria | Number(2) | N | Carga horária diária do funcionário | N | N | N | N |
| codSetor | Number(3) | Y | Código do setor onde o funcionário trabalha | N | S | N | N |
| codDepto | Number(3) | Y | Código do departamento onde o funcionário trabalha | N | S | N | N |
| dataEntrada | Date | N | Data de entrada do funcionário | N | N | N | N |
| dataSaida | Date | Y | Data de saída do funcionário | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FUNCIONARIOSTATUS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS SETORES | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codStatus | Number(3) | N | Código do status | S | N | N | N |
| status | Varchar2(20) | N | Nome do setor | N | N | N | N |
| matricula | Number(6) | N | Matricula do funcionario | N | S | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: EPI  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS MODELOS DOS EPI'S | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codEpi | Number(4) | N | Código do EPI | Y | N | N | N |
| nomeEpi | Varchar2(60) | N | Nome do EPI | N | N | N | N |
| origemEpi | Char(1) | N | Informa a origem do EPI | N | N | N | Y |
| ca | Number(12) | Y | Número de certificação do EPI. | N | N | N | N |
| validadeCa | date | Y | Data de validade da certificação do EPI | N | N | N | N |
| tamanho | Varchar2(4) | Y | Tamanho do EPI por número ou tipo | N | N | N | N |
| epiProtege | Varchar2(2000) | N | Descrição da proteção oferecida pelo EPI | N | N | N | N |
| descricaoEpi | Varchar2(2000) | Y | Campo livre para o usuário descrever o necessário sobre o EPI | N | N | N | N |
| QuantidadeMinima | Number(5) | N | Quantidade mínima em estoque do EPI | N | N | N | N |
| quantidade | Number(5) | N | Qauntidade em estoque do EPI | N | N | N | N |
| valorUnitário | Number(10,2) | N | Valor unitário do EPI | N | N | N | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABELA: TIPORISCO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS TIPOS DE RISCOS ENCONTRADOS | | | | | | |
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK |
| codTipoRisco | Number(4) | N | Código do tipo de risco | Y | N | N |
| nomeTipoRisco | Varchar2(60) | N | Nome do tipo do risco | N | N | N |
| descrição | Varchar2(2000) | Y | Campo livre para o usuário descrever o tipo do risco | N | N | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABELA: RISCOS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS RISCOS AMBIENTAIS | | | | | | |
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK |
| codRisco | Number(4) | N | Código de identificação do risco | Y | N | N |
| codTipoRisco | Number(4) | N | Código do tipo de risco informado na tabela TipoRisco | N | Y | N |
| efeitoSaude | Varchar2(2000) | N | Efeitos que o risco pode ocasionar a saúde | N | N | N |
| codEpi | Number(4) | N | Código do EPI utilizado para a proteção | N | Y | N |
| complemento | Varchar2(2000) | Y | Campo livre para anotações complementares do usuário | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: POSSUI  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS SETORES E DEPARTAMENTOS QUE POSSUEM RISCOS | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codPossuiRisco | Number(4) | N | Código da tabela possui risco | Y | N | N | N |
| codRisco | Number(4) | N | Associação das tabelas Risco com Possui Risco | N | Y | N | N |
| codSetor | Number(3) | Y | Associa o risco ao setor | N | Y | N | N |
| codDepto | Number(3) | Y | Associa o risco ao departamento | N | Y | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: PLANOCENTROCUSTO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DO PLANO DE CENTRO DE CUSTO | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codCentroCusto | Number(6) | N | Código do plano de centro de custos | Y | N | N | N |
| nomePlanoCentroCusto | Varchar2(30) | N | Nome do plano de centro de custo | N | N | N | N |
| validade | Date | N | Data de validade do centro de custo | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: CENTROCUSTODEPTO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS CENTROS DE CUSTO DOS DEPARTAMENTOS | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| centroCustoDepto | Number(5) | N | Código do centro de custo departamento | Y | N | N | N |
| codCentroCusto | Number(6) | N | Código do plano de centro de custos | N | Y | N | N |
| codDepto | Number(3) | N | Código do departamento | N | Y | Y | N |
| saldo | Number(10,2) | N | Saldo do centro de custo | N | N | N | N |
| validade | Date | N | Data de validade do centro de custo | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: CENTROCUSTOSETOR  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS CENROS DE CUSTO DOS SETORES | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| centroCustoSetor | Number(4) | N | Código do centro de custo setor | Y | N | N | N |
| centroCustoDepto | Number(5) | N | Código do centro de custo departamento | N | Y | N | N |
| codSetor | Number(3) | N | Código do setor | N | Y | Y | N |
| saldo | Number(10,2) | N | Saldo do centro de custo | N | N | N | N |
| validade | Date | N | Data de validade do centro de custo | N | N | N | N |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: ITENSADQUIRIDOS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE OS ITENS DAS AQUISIÇÕES DOS EPI’s | | | | | | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | | | PK | | FK | | UK | | CK |
| codAqui | Number(4) | N | Código do item adquirido | | | Y | | N | | N | | N |
| dataAquisicao | Date | N | Data de aquisição do EPI | | | N | | N | | N | | N |
| historico | Varchar2(3000) | Y | Informação digitada pelo usuário | | | N | | N | | N | | N |
| quantidade | Number(4) | N | Quantidade de EPI’s adquiridos | | | N | | N | | N | | N |
| valor | Number(10,2) | N | Valor do EPI | | | N | | N | | N | | N |
| centroCustoSetor | Number(4) | Y | Centro de custo do setor | | | N | | Y | | N | | N |
| centroCustoDepto | Number(5) | Y | Centro de custo departamento | | | N | | Y | | N | | N |
| codEpi | Number(4) | N | Código do EPI adquirido | | | N | | Y | | N | | N |
| matricula | Number(6) | N | Matricula do funcionário adquirente | | | N | | Y | | N | | N |
| TABELA: ITENSDEVOLVIDOS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE OS ITENS DAS DEVOLUÇÕES DOS EPI’s | | | | | | | | | | | | |
| ATRIBUTO | TIPO | NULO | | DESCRIÇÃO | PK | | FK | | UK | | CK | |
| codDev | Number(4) | N | | Código do item devolvido | Y | | N | | N | | N | |
| dataDevolucao | Date | N | | Data de devolução do EPI | N | | Y | | N | | N | |
| historico | Varchar2(3000) | Y | | Informação digitada pelo usuário | N | | N | | N | | N | |
| quantidade | Number(4) | N | | Quantidade de EPI’s devolvidos | N | | N | | N | | N | |
| valor | Number(10,2) | N | | Valor do EPI | N | | N | | N | | N | |
| centroCustoSetor | Number(4) | Y | | Centro de custo do setor | N | | Y | | N | | N | |
| centroCustoDepto | Number(5) | Y | | Centro de custo departamento | N | | Y | | N | | N | |
| codEpi | Number(4) | N | | Código do EPI adquirido | N | | Y | | N | | N | |
| matricula | Number(6) | N | | Matricula do Funcionário que está devolvendo o Epi | N | | Y | | N | | N | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: MOVIMENTA  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE A MOVIMENTAÇÃO DOS EPI’S | | | | | | | |
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codMov | Number(4) | N | Código de identificação do movimento | Y | N | N | N |
| codAqui | Number(4) | N | Código do item adquirido | N | Y | N | N |
| codDev | Number(4) | Y | Código do item devolvido | N | Y | N | N |
| id | Char(2) | N | Código da operação. “DB” “CR” | N | N | N | Y |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FORNECEDOR  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE OS FORNECEDORES DE EPI’s | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| cnpj | Number(18) | N | Cnpj do fornecedor | Y | N | N | N |
| nomeFornecedor | Varchar2(60) | N | Nome do fornecedor | N | N | N | N |
| fabricante | Char(1) | N | Identifica se o fabricante e nacional ou estrangeiro | N | N | N | Y |
| endereco | Ty\_Endereco | N | Endereço do fornecedor | N | N | N | N |
| contato | Ty\_Contato | N | Dados do contato com o fornecedor | N | N | N | N |
| dataCad | Date | N | Data do cadastro do fornecedor | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FORNECEDORSTATUS  DESCRIÇÃO: ARMAZENA O STATUS DO FORNECEDOR (ATIVO / INATIVO) | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codStatus | Number(3) | N | Código do status do Fornecedor | Y | N | N | N |
| status | Varchar2(20) | N | Status do Fornecedor | N | N | N | N |
| cnpj | Number(18) | N | Cnpj do fornecedor | N | Y | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FORNHSTATIVIDADE  DESCRIÇÃO: ARMAZENA O HISTORICO DO STATUS DO FORNECEDOR (ATIVO / INATIVO) | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codHstAtv | Number(3) | N | Código do histórico de atividade | Y | N | N | N |
| codStatus | Number(3) | N | Código do status do Fornecedor | N | Y | N | N |
| dataInatividade | Date | Y | Data em que o Fornecedor entrou na inatividade | N | N | N | N |
| dataVoltaAtividade | Date | Y | Data em que o Fornecedor entrou em atividade ou foi reativado | N | N | N | N |
| motivo | Varchar2(3000) | Y | Informações sobre a alteração de atividade / inatividade | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: EPIFORNECIDO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS EPI’S FORNECIDOS PELOS FORNECEDORES | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codEpiFornecido | Number(3) | N | Código do Epi fornecido | Y | N | N | N |
| dataInicio | Date | N | Data do inicio do fornecimento | N | N | N | N |
| codEpi | Number(4) | N | Código do EPI | N | Y | N | N |
| cnpj | Number(18) | N | Cnpj do Fornecedor | N | Y | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: FORNEPIHISTORICO  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DOS EPI’S QUE OS FORNECEDORES UM DIA FORNECERAM | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codHst | Number(3) | N | Código histórico do EPI fornecido | Y | N | N | N |
| codEpiFornecido | Number(3) | N | Código do Epi fornecido | N | Y | N | N |
| dataInicio | Date | N | Data do inicio do fornecimeno | N | N | N | N |
| dataFim | Date | Y | Data do fim do fornecimeno | N | N | N | N |
| historico | Varchar2(3000) | Y | Informações inseridas pelo usuário | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: COMPRAS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE AS SOLICITAÇÕES DE COMPRAS | | | | | | | |
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codCompra | Number(4) | N | Código de identificação da compra | Y | N | N | N |
| matricula | Number(6) | N | Matricula do funcionário solicitante da compra | N | Y | N | N |
| dataCompra | Date | Y | Data da solicitação ou compra do EPI | N | N | N | N |
| dataSolicitacao | Date | N | Data da solicitação de compras | N | N | N | N |
| motivoCompra | Varchar2(2000) | N | Motivo da compra do EPI | N | N | N | N |
| valorCompra | Number(10,2) | N | Valor da compra do EPI | N | N | N | N |
| statusCompra | Varchar2(12) | N | Status da compra do EPI | N | N | N | Y |
| numNota | Number(12) | Y | Número da nota fiscal | N | N | N | N |
| dataEntrega | Date | Y | Data da entrega da compra | N | N | N | N |
| observacoes | Varchar2(3000) | Y | Anotações do usuário | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: ITEMCOMPRAS  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES SOBRE AS SOLICITAÇÕES DE COMPRAS | | | | | | | |
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codItemCompra | Number(3) | N | Código do item da compra | Y | N | N | N |
| codCompra | Number(4) | N | Código de identificação da compra | Y | Y | N | N |
| codEpiFornecido | Number(3) | N | Código do Epi fornecido | N | Y | N | N |
| quantidade | Number(5) | N | Quantidade de EPI’S | N | N | N | N |
| descricao | Varchar2(1000) | Y | Descrição do Epi | N | N | N | N |
| valorUnitario | Number(10,2) | N | Valor unitário do item | N | N | N | N |
| valorTotal | Number(10,2) | N | Valor total do Item | N | N | N | N |
| observações | Varchar2(3000) | Y | Anotações do usuário | N | N | N | N |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA: BAIXAREPI  DESCRIÇÃO: ARMAZENA AS INFORMAÇÕES DAS BAIXAS DOS EPI’s | | | | | | | |
|
| ATRIBUTO | TIPO | NULO | DESCRIÇÃO | PK | FK | UK | CK |
| codBaixa | Number(4) | N | Código da baixa | Y | N | N | N |
| codEpi | Number(4) | N | Código do EPI a ser baixado | N | Y | N | N |
| matricula | number(6) | N | Matricula do funcionário que está efetuando a baixa | N | Y | N | N |
| nome | varchar2(60) | N | Nome do funcionário que está executando a baixa | N | N | N | N |
| codSetor | Number(3) | N | Código do setor que está efetuando a baixa | N | N | N | N |
| motivo | varchar2(25) | N | Motivo da baixa | N | N | N | Y |
| observacoes | varchar2(3000) | Y | Campo livre para a digitação do usuário | N | N | N | N |
| quantidade | Number(5) | N | Quandidade de EPI’s baixados | N | N | N | N |
| valorUnitario | Number(10,2) | N | Valor unitário do EPI | N | N | N | N |
| valorTotal | Number(10,2) | N | Soma dos valores do campo unitário | N | N | N | N |
| destino | varchar2(3000) | N | Destino do EPI | N | N | N | N |
| dataBaixa | date | N | Data da baixa do EPI | N | N | N | N |

**Mapeamento Diagrama de Classes X Tabelas de Banco de Dados**

|  |  |
| --- | --- |
| **Classes** | **Tabelas** |
| Funcionário | Funcionário |
| FuncionarioSatus | FuncionarioSatus |
| Departamento | Departamento |
| Setor | Setor |
| PlanoCentroCusto | PlanoCentroCusto |
| CentroCustoDepto | CentroCustoDepto |
| CentroCustoSetor | CentroCustoSetor |
| EPI | EPI |
| BaixarEpi | BaixarEpi |
| Compras | Compras |
| ItensCompras | ItensCompras |
| TipoRisco | TipoRisco |
| Riscos | Riscos |
| Possui | Possui |
| ItensAdquiridos | ItensAdquiridos |
| ItensDevolvidos | ItensDevolvidos |
| Movimenta | Movimenta |
| Fornecedor | Fornecedor |
| FornecedorStatus | FornecedorStatus |
| FornHstAtividade | FornHstAtividade |
| FonEpiHistorico | FonEpiHistorico |
| EpiFornecido | EpiFornecido |

**Nomeclatura para os objetos de banco de dados oracle**

Seguiremos as regras e as definições expostas abaixo para a formatação das nomenclaturas para tabelas, atributos, constraints e índices de banco de dados.

**Tabela**

Os nomes das tabelas sempre obedecerão às regras em maiúsculo evitando abreviações.

Tabela: DEPARTAMENTO

**Atributos**

Seguirá o padrão CamelCase, onde as palavras são descritas sem espaço e a primeira letra da palavra em minúsculo e as demais palavras a primeira letra deve ser em maiúsculo.

Atributo: codDepto – Código do departamento

**Restrições**

As restrições de integridade e as referencias seguirão os seguintes padrões de nomenclatura.

Chave Primária: PK\_Departamento

Chave Estrangeira: FK\_Departamento\_Setor

Chave única: UK\_CentroCusto

Check: CK\_Motivo

**Sequências Numéricas**

As sequências criadas com o objetivo de gerar números sequências para as chaves primárias deverão ser utilizadas somente por essas.

Caso haja necessidade de uma outro atributo utilizar números sequências deverá ser criada uma sequence só para esta.

Sequence: SQ\_ Departamento – para chaves primarias

Sequence: SQ\_Departamento\_nomeAtributo – para atributos que não sejam chave primária.

**CREATE USER PPRA IDENTIFIED BY root;**

**ALTER USER PPRA QUOTA UNLIMITED ON TBS\_PPRA\_DATA\_01;**

**ALTER USER PPRA QUOTA UNLIMITED ON TBS\_PPRA\_INDX\_01;**

**GRANT CONNECT, RESOURCE TO PPRA;**

**CREATE TABLE DEPARTAMENTO (**

**codDepto number(3) CONSTRAINT PK\_Departamento PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomeDepto varchar2(60) not null,**

**localDepto varchar2(60) not null,**

**matricula number(6)not null)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_departamento;**

**CREATE TABLE SETOR (**

**codSetor number(3) CONSTRAINT PK\_SETOR PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomeSetor varchar2(60) not null,**

**localSetor varchar2(60) not null,**

**matricula number(6) not null,**

**codDepto number(3) not null,**

**CONSTRAINT FK\_Setor\_CodDepto FOREIGN KEY(codDepto)**

**REFERENCES Departamento (codDepto)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_Setor\_CodDepto on Setor(codDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_setor;**

**CREATE TABLE FUNCIONARIO (**

**matricula number(6) CONSTRAINT PK\_Funcionario PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nome varchar2(60) not null,**

**cargo varchar2(50) not null,**

**funcao varchar2(50) not null,**

**ramal number(6),**

**email varchar2(30),**

**cargaHoraria number(2) not null,**

**codSetor number(3),**

**codDepto number(3),**

**dataEntrada date not null,**

**dataSaida date,**

**CONSTRAINT FK\_Funcionario\_CodSetor FOREIGN KEY(codSetor)**

**REFERENCES Setor (codSetor),**

**CONSTRAINT FK\_Funcionario\_CodDepto FOREIGN KEY(codDepto)**

**REFERENCES Departamento (codDepto)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_Funcionario\_CodSetor on FUNCIONARIO(codSetor)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Funcionario\_CodDepto on FUNCIONARIO(codDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE FUNCIONARIOSTATUS (**

**codStatus number(3) CONSTRAINT PK\_FuncStatus PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**status varchar2(20),**

**matricula number(6),**

**CONSTRAINT FK\_FuncStatus FOREIGN KEY(matricula)**

**REFERENCES Funcionario (matricula)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_FuncStatus on FUNCIONARIOSTATUS(matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_funcionariostatus;**

**alter table departamento**

**add constraint FK\_Depart\_Func Foreign key (matricula)**

**references Funcionario(matricula);**

**create index FK\_Depart\_Func on departamento(matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**alter table setor**

**add constraint FK\_Setor Foreign Key (matricula)**

**references Funcionario(matricula);**

**create index FK\_Setor on setor(matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TYPE TY\_ENDERECO AS OBJECT (**

**rua varchar2(60),**

**numero number(6),**

**complemento varchar2(12),**

**Bairro varchar2(30),**

**Cep varchar2(9),**

**cidade varchar2(30),**

**uf char(2));**

**CREATE TYPE TY\_CONTATO AS OBJECT (**

**tel1 varchar2(9),**

**tel2 varchar2(9),**

**ramal1 number(4),**

**ramal2 number(4),**

**email1 varchar2(40),**

**email2 varchar2(40));**

**CREATE TABLE EPI (**

**codEpi number(4) constraint PK\_EPI PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomeEpi varchar2(60) not null,**

**origemEpi char(1) not null,**

**CA number(12) constraint UK\_CA unique**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**validadeCA date,**

**tamanho varchar2(4),**

**epiProtege varchar2(2000) not null,**

**descricaoEpi varchar2(2000),**

**quantMinima number(5) not null,**

**quantidade number(5) not null,**

**valorUnitario number(10,2) not null,**

**constraint CNK\_EPI\_QTD check (quantidade >= 0),**

**constraint CNK\_origemEpi check (origemEpi = 'E' or origemEpi = 'N')**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_epi;**

**CREATE TABLE TipoRisco (**

**codTipoRisco number(4) constraint PK\_TipoRisco PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomeTipoRisco varchar2(60) not null,**

**descricao varchar2(2000))**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_tiporisco;**

**CREATE TABLE Riscos**

**(codRisco number(4) constraint PK\_Riscos PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codTipoRisco number(4) not null,**

**efeitoSaude varchar2(2000) not null,**

**codEpi number(4) not null,**

**complemento varchar2(2000),**

**constraint FK\_Riscos\_EPI FOREIGN KEY (codEpi) REFERENCES EPI(codEpi),**

**constraint FK\_Riscos\_TipoRisco FOREIGN KEY (codTipoRisco) references TipoRisco(codTipoRisco)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_Riscos\_EPI on Riscos(codEpi)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Riscos\_TipoRisco on Riscos(codTipoRisco)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_riscos;**

**CREATE TABLE PossuiRisco (**

**codPossuiRisco number(4) constraint PK\_PossuiRisco PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codRisco number(4) not null,**

**codSetor number(3),**

**codDepto number(3),**

**constraint FK\_PossuiRisco\_Risco FOREIGN KEY(codRisco) REFERENCES Riscos(codRisco),**

**constraint FK\_PossuiRisco\_Setor FOREIGN KEY(codSetor) REFERENCES Setor(codSetor),**

**constraint FK\_PossuiRisco\_Depto FOREIGN KEY(codDepto) REFERENCES Departamento(codDepto)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_PossuiRisco\_Risco on PossuiRisco(codRisco)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_PossuiRisco\_Setor on PossuiRisco(codSetor)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_PossuiRisco\_Depto on PossuiRisco(codDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_possuirisco;**

**CREATE TABLE PlanoCentroCusto (**

**codCentroCusto number(6) constraint PK\_PlanoCentroCusto PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomePlanoCentroCusto varchar2(30) not null,**

**validade date not null**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**CREATE TABLE CentroCustoDepto (**

**centroCustoDepto number(5) constraint PK\_CentroCustoDepto PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codCentroCusto number(6) not null,**

**codDepto number(3) not null,**

**saldo number(10,2) not null,**

**validade date not null,**

**constraint FK\_CtrCustoDepto\_PlCtrCusto FOREIGN KEY (codCentroCusto)**

**REFERENCES PlanoCentroCusto(codCentroCusto),**

**constraint FK\_CtrCustoDepto\_Depto FOREIGN KEY (codDepto)**

**REFERENCES Departamento(codDepto)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_CtrCustoDepto\_PlCtrCusto on CentroCustoDepto(codCentroCusto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_CtrCustoDepto\_Depto on CentroCustoDepto(codDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE CentroCustoSetor (**

**centroCustoSetor number(4) constraint PK\_CentroCustoSetor PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**centroCustoDepto number(5) not null,**

**codSetor number(3) not null,**

**saldo number(10,2) not null,**

**validade date not null,**

**constraint FK\_CtrCustoSetor\_CtrCDepto FOREIGN KEY (centroCustoDepto)**

**REFERENCES CentroCustoDepto(centroCustoDepto),**

**constraint FK\_CtrCustoSetor\_Setor FOREIGN KEY (codSetor)**

**REFERENCES Setor(codSetor)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_CtrCustoSetor\_CtrCDepto on CentroCustoSetor(centroCustoDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_CtrCustoSetor\_Setor on CentroCustoSetor(codSetor)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE ItensAdquiridos (**

**codAqui number(4) constraint PK\_ItensAdquiridos PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**dataAquisicao date not null,**

**historico varchar2(3000),**

**quantidade number(4) not null,**

**valor number(10,2) not null,**

**centroCustoSetor number(4),**

**centroCustoDepto number(5),**

**codEpi number(4) not null,**

**matricula number(6) not null,**

**constraint FK\_Itens\_CtrCustoSetor Foreign Key (centroCustoSetor)**

**references CentroCustoSetor (centroCustoSetor),**

**constraint FK\_Itens\_CtrCustoDepto Foreign Key (centroCustoDepto)**

**references CentroCustoDepto (centroCustoDepto),**

**constraint FK\_ItensAdquiridos\_EPI Foreign Key (codEpi)**

**references EPI (codEpi),**

**constraint FK\_Iten\_Func Foreign Key (matricula)**

**references Funcionario (matricula)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_Itens\_CtrCustoSetor on ItensAdquiridos(centroCustoSetor)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Itens\_CtrCustoDepto on ItensAdquiridos(centroCustoDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_ItensAdquiridos\_EPI on ItensAdquiridos(codEpi)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Iten\_Func on ItensAdquiridos(matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_itensadquiridos;**

**CREATE TABLE ItensDevolvidos (**

**codDev number(4) constraint PK\_ItensDevolvidos PRIMARY KEY**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**dataDevolucao date not null,**

**historico varchar2(3000),**

**quantidade number(4) not null,**

**valor number(10,2) not null,**

**centroCustoSetor number(4),**

**centroCustoDepto number(5),**

**codEpi number(4) not null,**

**matricula number(6) not null,**

**constraint FK\_ItensDev\_CtrCustoSetor Foreign Key (centroCustoSetor)**

**references CentroCustoSetor (centroCustoSetor),**

**constraint FK\_ItensDev\_CtrCustoDepto Foreign Key (centroCustoDepto)**

**references CentroCustoDepto (centroCustoDepto),**

**constraint FK\_Itens\_DevolucaoEpi Foreign Key (codEpi)**

**references EPI (codEpi),**

**constraint FK\_Iten\_Dev\_Func Foreign Key (matricula)**

**references Funcionario (matricula)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create index FK\_ItensDev\_CtrCustoSetor on ItensDevolvidos(centroCustoSetor)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_ItensDev\_CtrCustoDepto on ItensDevolvidos(centroCustoDepto)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Itens\_DevolucaoEpi on ItensDevolvidos(codEpi)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Iten\_Dev\_Func on ItensDevolvidos(matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create sequence sq\_itensdevolvidos;**

**CREATE TABLE MOVIMENTA (**

**codMov number(4) constraint PK\_Movimenta Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codAqui number(4) not null,**

**codDev number(4),**

**id char(2) not null,**

**constraint CK\_ID Check (id = 'CR' or id = 'DB'),**

**constraint FK\_Mov\_ItensAdq Foreign Key (codAqui)**

**references ItensAdquiridos(codAqui),**

**constraint FK\_Mov\_ItensDev Foreign Key (codDev)**

**references ItensDevolvidos(codDev)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_movimenta;**

**create index FK\_Mov\_ItensAdq on MOVIMENTA(codAqui)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_Mov\_ItensDev on MOVIMENTA(codDev)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE Fornecedor (**

**cnpj number(18) constraint PK\_Fornecedor Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**nomeFornecedor varchar2(60) not null,**

**fabricante char(1) not null,**

**endereco ty\_endereco not null,**

**contato ty\_contato not null,**

**constraint CNK\_Fabricante check (Fabricante = 'S' or Fabricante = 'N')**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**CREATE TABLE FornecedorStatus (**

**codStatus number(3) constraint PK\_FornecedorStatus Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**status varchar2(20) not null,**

**cnpj number(18) not null,**

**constraint FK\_FornStatus\_Forn Foreign Key (cnpj)**

**references Fornecedor(cnpj)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_fornecedorstatus;**

**create index FK\_FornStatus\_Forn on FornecedorStatus(cnpj)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE FornHstAtividade (**

**codHstAtv number(3) constraint PK\_FornHstAtividade Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codStatus number(3) not null,**

**dataInatividade date,**

**dataVoltaAtividade date,**

**motivo varchar2(3000)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_fornhstatividade;**

**CREATE TABLE EpiFornecido (**

**codEpiFornecido number(3) constraint PK\_EpiFornecido Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**dataInicio date not null,**

**codEpi number(4) not null,**

**cnpj number(18) not null,**

**constraint FK\_EpiForn\_Epi Foreign Key (codEpi)**

**references Epi(codEpi),**

**constraint FK\_EpiForn\_Forn Foreign Key (cnpj)**

**references Fornecedor(cnpj)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_codepifornecido;**

**create index FK\_EpiForn\_Epi on EpiFornecido(codEpi)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_EpiForn\_Forn on EpiFornecido(cnpj)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE FornEpiHistorico (**

**codHst number(3) constraint PK\_FornEpiHistorico Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codEpiFornecido number(3) not null,**

**dataInicio date not null,**

**dataFim date,**

**historico varchar2(3000),**

**constraint FK\_FornEpiHist\_EpiForn Foreign Key (codEpiFornecido)**

**references EpiFornecido(codEpiFornecido)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_fornepihistorico;**

**create index FK\_FornEpiHist\_EpiForn on FornEpiHistorico(codEpiFornecido)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE Compras (**

**codCompra number(4) constraint PK\_Compras Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**matricula number(6) not null,**

**motivoCompra varchar2(2000) not null,**

**dataSolicitacao date not null,**

**valorCompra number(10,2) not null,**

**dataCompra date,**

**statusCompra varchar2(12) not null,**

**numNota number(12),**

**dataEntrega date,**

**observacoes varchar2(3000),**

**constraint CNK\_Compras check(statusCompra = 'PENDENTE' or statusCompra = 'CANCELADO'**

**or statusCompra = 'SUSPENSA' or statusCompra = 'DEVOLVIDO'**

**or statusCompra = 'REJEITADO' or statusCompra = 'REALIZADO'),**

**constraint FK\_Compras\_Func Foreign Key (matricula)**

**references Funcionario(matricula)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_compras;**

**create index FK\_Compras\_Func on Compras (matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE ItensCompras (**

**codItemCompra number(3),**

**codCompra number(4),**

**codEpiFornecido number(3) not null,**

**quantidade number(5) not null,**

**valorUnitario number(10,2) not null,**

**valorTotal number(10,2) not null,**

**constraint FK\_ItensCompras\_Compras Foreign Key (codCompra)**

**references Compras (codCompra),**

**constraint FK\_ItensCompras\_EpiForn Foreign Key (codEpiFornecido)**

**references EpiFornecido (codEpiFornecido),**

**constraint PK\_ItensCompras Primary Key (codItemCompra, codCompra)**

**using index**

**tablespace TBS\_PPRA\_INDX\_01**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_itenscompras;**

**create index FK\_ItensCompras\_Compras on ItensCompras (codCompra)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_ItensCompras\_EpiForn on ItensCompras (codEpiFornecido)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index IX\_codItemCompra on ItensCompras (codItemCompra)**

**tablespace TBS\_PPRA\_INDX\_01;**

**CREATE TABLE BaixarEpi (**

**codBaixa number(4) constraint PK\_BaixarEpi Primary Key**

**using index**

**tablespace TBS\_PPRA\_INDX\_01,**

**codEpi number(4) not null,**

**matricula number(6) not null,**

**nome varchar2(60) not null,**

**codSetor number(3) not null,**

**motivo varchar2(25) not null,**

**observacoes varchar2(3000),**

**quantidade number(5) not null,**

**valorUnitatio number(10,2) not null,**

**valorTotal number(10,2) not null,**

**destino varchar2(3000) not null,**

**dataBaixa date not null,**

**constraints CNK\_Motivo check (**

**motivo = 'Validade Expirada' or**

**motivo = 'CA Vencido' or**

**motivo = 'Degradado' or**

**motivo = 'Má Conservação' or**

**motivo = 'Danificado' or**

**motivo = 'Perdido' or**

**motivo = 'Extraviado' or**

**motivo = 'Furtado' or**

**motivo = 'Roubado'**

**),**

**constraints FK\_BaixarEpi\_EPI Foreign Key (codEpi)**

**references EPI(codEpi),**

**constraints FK\_BaixarEpi\_Func Foreign Key (matricula)**

**references Funcionario(matricula)**

**)**

**tablespace TBS\_PPRA\_DATA\_01;**

**create sequence sq\_baixarepi;**

**create index FK\_BaixarEpi\_EPI on BaixarEpi (codEpi)**

**tablespace TBS\_PPRA\_INDX\_01;**

**create index FK\_BaixarEpi\_Func on BaixarEpi (matricula)**

**tablespace TBS\_PPRA\_INDX\_01;**

**Conceitos do SGBD Oracle**

O Sistema de Gerenciamento de Banco de Dados Oracle (SGBD) selecionado para este projeto será a versão 11gr2 Standard Edition One. Esta é uma versão mais simples do SGBD Oracle trabalhando em um único servidor com dois socketes, e suporta os principais Sistemas Operacionais de Marcado, sendo eles: Windows Server, Linux e Unix.

O SGBD Oracle implementa os dois modelos hoje utilizados no mercado de tecnologia que são estes o Modelo Realacional e o Modelo Objeto-Relacional.

“**Relational Database Management System (RDBMS)**

O modelo relacional é a base para um sistema de gerenciamento de banco de dados relacional (RDBMS). Essencialmente, um RDBMS move dados em um banco de dados e armazena os dados, e recuperá-lo de modo que ele pode ser manipulado por aplicações.   
  
 Oracle Database é um RDBMS que implementa recursos orientados a objetos como tipos definidos pelo usuário, herança e polimorfismo é chamado de  
sistema de gestão de banco de dados objeto-relacional (ORDBMS). Oracle Database estendeu o modelo relacional para o modelo de objeto-relacional, tornando possível armazenar complexos modelos de negócios em um banco de dados relacional.”

*(fonte: Oracle® Database Concepts 11g Release 2 (11.2) E25789-01)*

**Modelo Relacional**

É atualmente o mais utilizado hoje no mercado de banco de dados e suportado por praticamente todas as ferramentas SGBD e de fácil alocação de mão de obra profissional.

Têm como características principais representar os dados com uma coleção de tabelas que se relacionam entre si cada uma com o seu nome no qual cada tabela possui uma coleção de atributos com os seus tipos de dados.

Adotam a linguagem SQL.

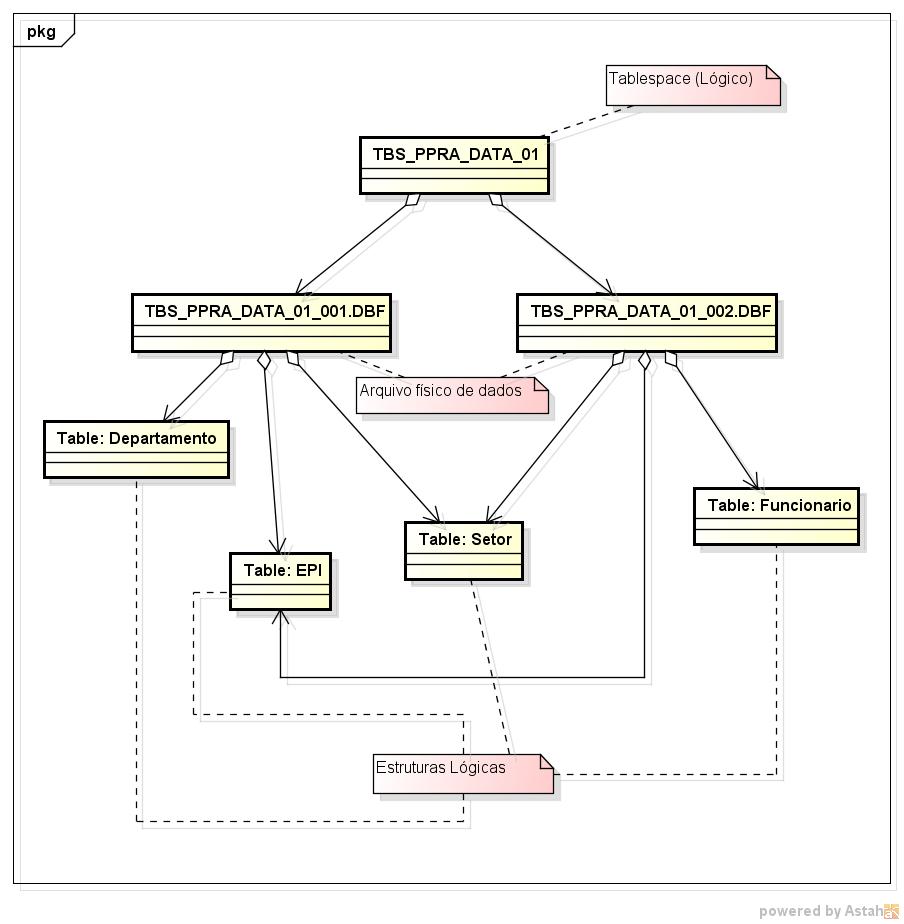
**Modelo Objeto-Relacional**

É uma extensão do modelo relacional, que implementa conceitos da orientação a objetos como herança e polimorfismo, tipos de objetos e adota a linguagem sql estendida (sql 2003).

**Estrutura de Armazenamento Físico**

**“**Uma característica de um RDBMS é a independência das estruturas lógicas de dados, tais como tabelas, visões, e os índices de estruturas de armazenamento físico. Por causa das estruturas físicas e lógicas serem separadas, você pode gerenciar o armazenamento físico de dados sem afetar o acesso a estruturas lógicas. Por exemplo, renomear um arquivo de banco de dados não renomear as tabelas armazenados na mesma. “ *(fonte: Oracle® Database Concepts 11g Release 2 (11.2) E25789-01)*

O Oracle trabalha com o conceito de tablespace. As tablespaces são lógicas enquanto os arquivos de dados ligados a essas tablespaces são físicos. As tabelas e índices são estruturas lógicas armazenadas nos datafiles (estrutura física).

Figura 1: Estrutura da tablespace TBS\_PPRA\_DATA\_01

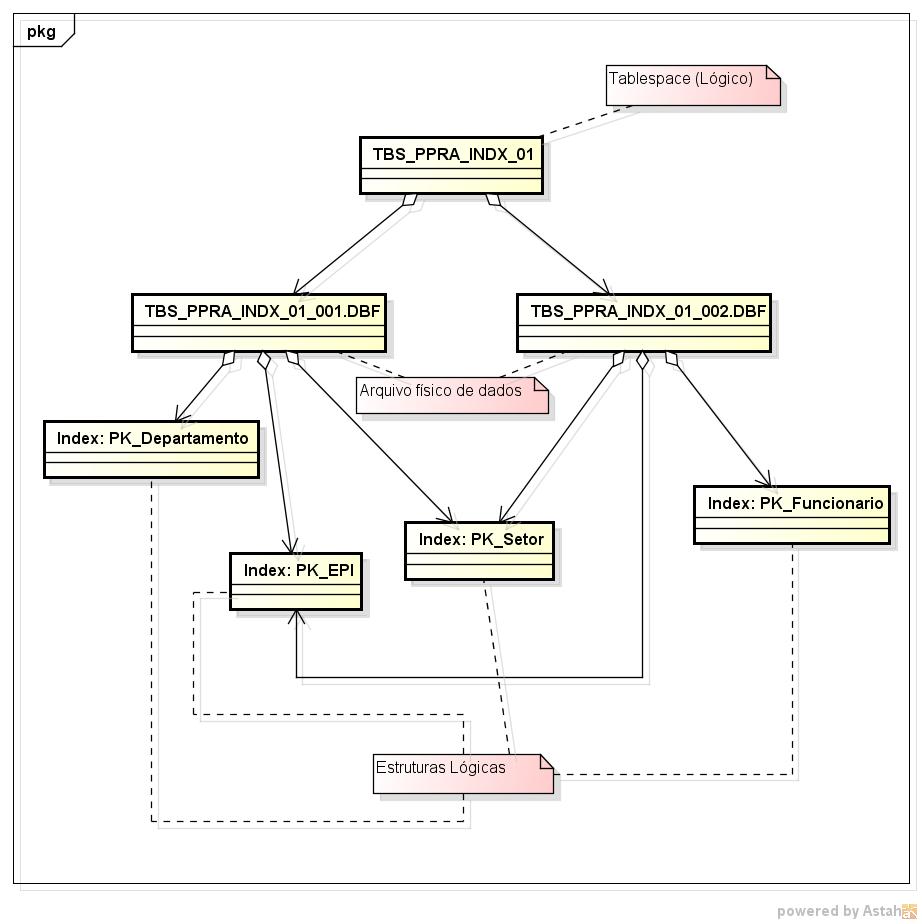


Figura 2: Estrutura da tablespace TBS\_PPRA\_INDX\_01

**Advanced Storage Manager**

Consiste em criar um grupo de discos em uma estrutura lógica. O ASM se encarrega de distribuir os blocos físicos dos datafiles entre os discos pertencentes ao grupo para balanceamento de I/O.

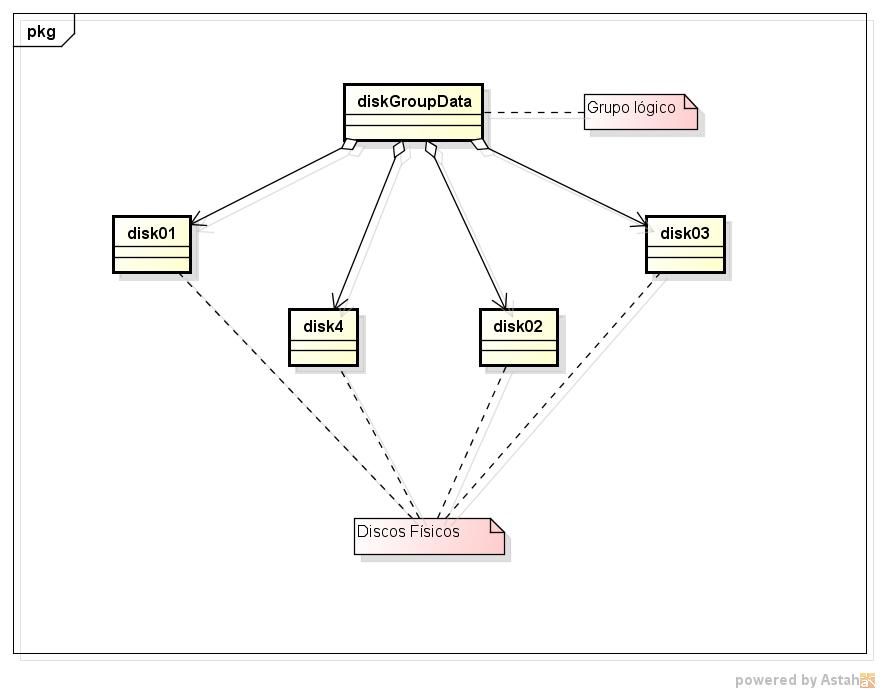


Figura 3: Estrutura Asm para dados

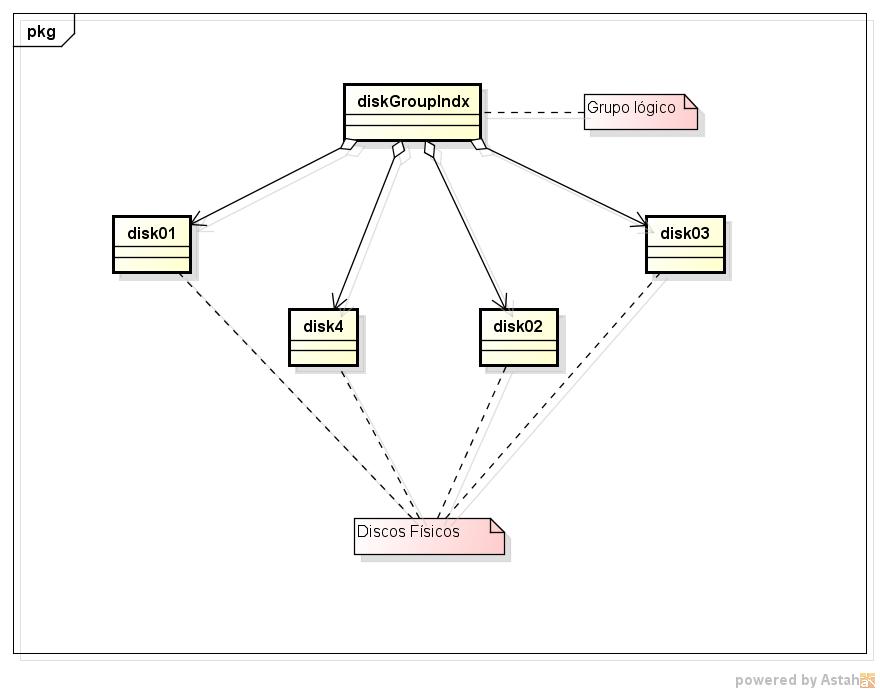


Figura 3: Estrutura Asm para índices

Não é aconselhável misturarmos tabelas e índices na mesma tablespace, e nem datafiles de dados nos mesmos disco ou diskgroup de índices.

O mesmo raciocínio aplica-se para tabelas temporárias que devem estar em tablespaces temporárias.

**RAC**

Pode ser aplicado quando se tem mais de um servidor, para balanceamento de carga e prevenção de falha de hardware excetuando falha no storage.

**Manutenção de Banco de Dados**

**Estatísticas das Tabelas e Índices**

**“**Estatísticas do otimizador são uma coleção de dados que descrevem os detalhes sobre o banco de dados e os objetos no banco de dados. As estatísticas dão uma imagem estatística correta de armazenamento de dados e distribuição utilizável pelo otimizador ao avaliar os caminhos de acesso.” *(fonte: Oracle® Database Concepts 11g Release 2 (11.2) E25789-01)*

O seguinte comando devem ser executados mensalmente para garantir a boa performance do banco de dados.

EXEC DBMS\_STATS.GATHER\_DATABASE\_STATS(estimate\_percent => 70, block\_sample => TRUE, method\_opt =>'FOR ALL COLUMNS SIZE AUTO',degree =>2,granularity =>'ALL',cascade => TRUE,options =>'GATHER',gather\_sys => TRUE);

O comando acima pode ser schedulado para o dia primeiro de cada dia às 20:00 horas.

BEGIN

-- Job defined entirely by the CREATE JOB procedure.

DBMS\_SCHEDULER.create\_job (

job\_name => 'Analyze\_Database',

job\_type => 'PLSQL\_BLOCK',

job\_action => 'BEGIN DBMS\_STATS.GATHER\_DATABASE\_STATS(

estimate\_percent => 70 ,

block\_sample => TRUE,

method\_opt =>''FOR ALL COLUMNS SIZE AUTO'',

degree =>2, granularity =>''ALL'',

cascade => TRUE,

options =>''GATHER'',

gather\_sys => TRUE); END;',

start\_date => SYSTIMESTAMP,

repeat\_interval => 'FREQ=MONTHLY; BYMONTHDAY=1; BYHOUR=20',

end\_date => NULL,

enabled => TRUE,

comments => 'Job defined entirely by the CREATE JOB procedure.');

END;

**Performance**

Para garantia de boa performance indicamos a configuração dos seguintes parâmetros de memória.

SGA\_MAX\_SIZE → parâmetro de inicialização.

SGA\_TARGET → redimensiona automaticamente o pool de memória otimizando automaticamente o mesmo.

Os parâmetros acima uma vez ativados permite o gerenciamento automático de memória por parte do Banco de dados Oracle ajustando automaticamente a memória para as principais estruturas de memória, são elas:

DATABASE BUFFER CACHE (DEFAULT POOL) → espaço de memória que armazena dados a serem consultados ou modificados pelos usuários. Evita o excesso de leitura em disco (I/O).

SHARED POOL → armazena as instruções SQL que podem ser reutilizadas, informações de dicionário de dados e procedimentos que são o código executável.

LARGE POOL → área de grande alocação de memória para as operações de backup e restauração e os processos de servidor de I/O.

JAVA POOL → sessão específica do código java e dados dentro da máquina virtual java.

STREAMS POOL → área de memória utilizado pelo oracle streams.

**Política de Backup**

**Backup Lógico (DataPunp)**

Data Pump Export é um utilitário para descarregamento de dados e metadados em um conjunto de arquivos do sistema operacional chamado um conjunto de arquivos de despejo.  
 Os dumps files só podem ser importados pelo utilitário de importação Data Pump e pode ser importado no mesmo sistema, ou pode ser movido para outro sistema e carregado lá.

**Backup Lógico Full**

O Data Pump é baseado em servidor ao invês de cliente, por causa desta condição e necessário especificar os paths de diretório conforme exemplo abaixo.

SQL> CREATE DIRECTORY dpump\_dir1 AS '/usr/apps/datafiles';

O seguinte comando abaixo deve ser executado diariamente no prompt de comando do sistema operacional.

expdp xe FULL=YES DUMPFILE=dpump\_dir1:full1%U.dmp,

FILESIZE=2G PARALLEL=3 LOGFILE=dpump\_dir1:expfull.log JOB\_NAME=expfull

**Backup Físico (Rman)**

O Recover Manager é a ferramenta de backup fornecido juntamente com o SGBD Oracle para a realização de backup físico full ou incremental. O Recover Manager copia somente os blocos utilizados pelo SGBD. Esta cópia pode ser direcionada para disco ou fita. O Recover Manager possui interface com várias ferramentas de gerenciamento de backup do mercado de TI.

Para este projeto utilizaremos o backup offline full diário com retenção de uma semana por se tratar de um banco de dados previsionalmente pequeno.

O backup offline full é conhecido como backup consistente, já que o banco de dados deve

ser primeiramente posto no estado offline e posteriormente no estado mount para início do backup.

O banco de estado em estado mount fica inacessível para os usuários. Somente no estado open o banco de dados está disponível para o usuário.

Colocando o banco de dados em modo archivelog

SQL\*Plus: Release 11.2.0.2.0 Production on Ter Mai 29 10:34:27 2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> conn sys / as sysdba

Informe a senha:

Conectado.

SQL> shutdown immediate

Banco de dados fechado.

Banco de dados desmontado.

InstÔncia ORACLE desativada.

SQL>

SQL> startup mount

InstÔncia ORACLE iniciada.

Total System Global Area 535662592 bytes

Fixed Size 1384760 bytes

Variable Size 331353800 bytes

Database Buffers 197132288 bytes

Redo Buffers 5791744 bytes

Banco de dados montado.

SQL>

SQL> alter database archivelog;

Banco de dados alterado.

Política de retenção de backup

Ex.: RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 days;

Script de Backup do Banco de Dados (modo archivelog)

run {

sql “shutdown immediate”;

sql “startup mount”;

ALLOCATE CHANNEL T1 TYPE DISK;

BACKUP DATABASE FORMAT ‘C:\BACKUP\DATABASE\bkp\_off\_full\_DB\_%s\_%p\_%t'.bkp;

BACKUP ARCHIVELOG ALL FORMAT ‘C:\BACKUP\ARCHIVELOG\bkp\_ARQ\_%s\_%p\_%t.bkp' DELETE ALL INPUT;

BACKUP SPFILE FORMAT ‘C:\BACKUP\SPFILE\bkp\_SPFILE\_%s\_%p\_%t.bkp';

BACKUP CURRENT CONTROLFILE FORMAT ‘C:\BACKUP\CONTROLFILE\bkp\_CTRL\_%s\_%p\_%t.bkp';

sql "ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS ''C:\BACKUP\CONTROLFILE\bkp\_CTRL.sql'' reuse";

sql “ALTER DATABASE OPEN”;

}