Ministério da Educação
Secretaria da Educação Profissional e Tecnológica
Instituto Federal Catarinense
Campus Videira

LUCAS DOS SANTOS CHEROBIM

JOHN VICTOR GOMES

WILLIAM MOREIRA VALTER

PROJETO IRINEU VERSÃO 2:

BD DO SISTEMA PARA CLASSIFICAÇÃO DOS CANDIDATOS INSCRITOS NOS PROCESSOS SELETIVOS

Videira - SC

2017

LUCAS DOS SANTOS CHEROBIM

JOHN VICTOR GOMES

WILLIAM MOREIRA VALTER

PROJETO IRINEU VERSÃO 2:

BD DO SISTEMA PARA CLASSIFICAÇÃO DOS CANDIDATOS INSCRITOS NOS PROCESSOS SELETIVOS

Trabalho apresentado como exigência parcial para o curso de graduação em ciência da computação do Instituto Federal de Educação Ciência e Tecnologia Catarinense – Câmpus Videira para obtenção da nota parcial em Banco de dados 2.

Professor: Tiago Heineck.

Videira – SC

2017

1 DESCRIÇÃO DO MUNDO REAL

Antes de ingressar em uma instituição é necessário avaliar através de uma prova de conhecimentos o nível de cada candidato, classificando-os de acordo com sua performance. O processo seletivo leva em consideração não somente o conhecimento de cada candidato, mas sua situação como pessoa, visto que alguns podem ter mais dificuldades que outros.

Dito isso, foi realizada a proposta do desenvolvimento de um programa de computador para classificação dos candidatos inscritos nos processos seletivos do IFC.

Ao que compete o projeto, nesta fase será construído o banco de dados bem como os fatores que se fazem necessários para atribuição de notas ao mesmo.

2 DESCRIÇÃO DOS OBJETIVOS DA APLICAÇÃO

O objetivo da aplicação é possibilitar a classificação do certame (concurso em questão) para ingresso nos cursos do IFC.

A entrada principal do programa é um arquivo que contém todos os candidatos que realizaram a prova e tiveram pontuação mínima para ser aprovado. Esse arquivo conterá o número de inscrição do candidato, o *campus* e curso pretendido. a nota final e a cota a qual ele pertence. O programa deverá, a partir das cotas cadastradas para cada curso, classificar os candidatos. Caso sobrem vagas em algumas cotas, o programa deverá buscar, para cada vaga. a próxima cota cadastrada. E conforme legislação vigente.

2.1 REQUISITOS

O programa deverá ter os seguintes requisitos obrigatórios:

a) Permitir o cadastro de cotas, incluindo os campos código e descrição (as cotas atuais podem ser visualizadas nos anexos I e II.
b) Permitir o cadastro de oferta de cursos, incluindo os campos código, descrição e campus;
c) Permitir o cadastro de oferta de vagas por curso/cota, com os campos código de cota, código do curso e número de vagas:
d) Realizar a classificação dos aprovados em cada cota. a partir do arquivo da listagem de candidatos e do cadastro de oferta de vagas por curso/cota:
e) Realizar a classificação dos aprovados em cada cota. considerando as vagas de cotas não preenchidas e os critérios de realocação de cotas, conforme anexos I e II.

3 STORY USERS CARDS

3.1 CADASTRO-PESSOA

Como gestor quero que o sistema possua um cadastro de pessoas para que elas possam se cadastrar com seus dados pessoais e endereço

Dados importantes:

Nome completo

Data de Nascimento

R.G. 34.235.358-75

emissor do R.G.

CPF

Passaporte (se for estrangeiro)

Endereços - podendo ser comercial, residencial ou outro
(rua, número, bairro, cidade, estado|sigla e nome, país|sigla e nome)

Telefones - podendo ser residencial, celular, comercial ou outro

Nome do Pai

Nome da Mãe

escolaridade
(aqui a pessoa informará sua escolaridade, também deverá dizer se ela foi feito inteiramente em escola pública, inteiramente em escola particular ou em escola particular com bolsa integral) -> aqui vai de cada um !!!

usuario

senha

e-mail

data do cadastro

data da última atualização

ativo

3.2 PREFERÊNCIAS-DA-PESSOA

Como usuário quero escolher que tipos de notificações quero receber para que receba e-mail somente do que considero importante

O que é importante:

Tipos de notificações:

novo processo aberto,

inscrição validada,

inscrição aceita,

última semana para inscrição,

último dia para inscrição

O usuário pode escolher qual desse tipo vai receber

3.3 NECESSIDADES-ESPECIAIS

Como deficiente físico quero poder identificar o IFC sobre minhas deficiências para que possa ter acesso adequada a realização da prova Questões importantes: Existem diversos tipos de deficiências, algumas são permanentes, a pessoa terá aquela condição em todo o processo que ela realizar Algumas outras podem ser temporárias, por exemplo: um braço quebrado. Essas a pessoa poderá informar a cada novo processo que ela realizar

3.4 NATURALIDADE-NACIONALIDADE

Como gestor, quero saber a naturalidade e a nacionalidade da pessoa, para fins de estatísticas.

naturalidade é a cidade onde a pessoa nasceu

nacionalidade é o país onde a pessoa nasceu

3.5 PROCESSO-SELETIVO

Como gestor, quero poder cadastrar um processo seletivo no sistema para que seja ofertado ao público interessado em ser aluno do IFC.

Informações importantes

- O processo seletivo deve ter uma data de início e uma de fim do período de inscrição, o sistema deve bloquear inscrições fora deste período

- O processo seletivo pode ser "exame de classificação" para ensino médio, "vestibular" para ensino superior e "concurso" para contratação de funcionários

- Deve ter uma data de vencimento do boleto de inscrição, para que seja emitido boleto com este vencimento, quando não for gratuito

- Alguns processos seletivos possuem taxa de inscrição, nesse caso o valor da taxa deve ser armazenado

- Alguns tipos de candidatos podem ter isenção da taxa de inscrição mediante solicitação

- O sistema deverá controlar os usuários que pagaram a taxa de inscrição e a data que foi paga

- Todo processo seletivo deverá ter uma comissão que é composta por pessoas, uma pessoa pode participar de vários processos seletivos como comissão,

isso é importante porque pessoas da comissão tem acesso a funcionalidades específicas do sistema

3.6 INSCRIÇÃO

Como candidato quero me inscrever em um processo seletivo para poder participar no dia da prova

Regras:

Um candidato deve escolher um curso como primeira opção

Um candidato pode escolher um curso como segunda opção

Um candidato pode escolher uma cota para concorrer a vaga pretendida, as cotas estão todas especificadas no arquivo do edital

Um candidato/pessoa pode se inscrever em quantos processos seletivos ele quiser

A inscrição deve guardar a data e hora que foi realizada

3.7 PAGAMENTO-DE-TAXA

Como financeiro quero saber quando e quem pagou a taxa de inscrição para ter controle dos usuários que podem ser homologados

Regras:

O sistema deve armazenar a data que cada pessoa pagou por cada uma de suas inscrições

3.8 ISENÇÃO-DE-TAXA

Como assistente social quero saber quem solicitou isenção de taxa para que possa analisar e assim homologar as inscrições por isenção

Regras:

O candidato deverá solicitar isenção por formulário próprio, informando o motivo e em qual processo que está participando está solicitando a isenção

O sistema deve armazenar se a solicitação foi aceita ou não e qual usuário membro da comissão fez o aceite

4 SGBD

Um banco de dados é uma coleção de tabelas relacionadas que são geralmente integradas, vinculadas ou referenciadas a um outro. A vantagem de um banco de dados é que os dados e registros contidos em tabelas diferentes podem ser facilmente organizadas e recuperadas utilizando software de gestão especializado chamado de sistema gerenciador de banco de dados (SGBD) ou gerente de banco de dados.

Fundamentos SGBD

Um sistema de gerenciamento de banco de dados é um conjunto de programas de software que permite aos usuários criar, editar, atualizar, armazenar e recuperar dados em tabelas de banco de dados. Dados em um banco de dados podem ser acrescentados, apagados, alterados, classificados usando um SGBD. Se você fosse um empregado em uma grande organização, a informação sobre você provavelmente seria armazenadas em diferentes

tabelas que estão ligados entre si. Por referência cruzada dessas tabelas, alguém poderia mudar o endereço de uma pessoa em uma tabela e ela seria automaticamente refletida para todas as outras tabelas.

Características Desejáveis numa Database

>Controle de Redundância;
>Compartilhamento de Dados;
>Controle de Acesso aos Dados;
>Múltiplas Interfaces;
>Representação de associações complexas;
>Garantia de restrições de Integridade;
>Recuperação de falhas.

SGBDs são comumente usados para gerenciar:

>Sócios e listas de discussão de subscrição
>Informação contábil e contabilidade
>Os dados obtidos a partir de pesquisa científica
>Informações de clientes
>Informações de inventário
>Registros pessoais
>Informações da biblioteca

As vantagens de um SGBD

Maior disponibilidade: Uma das principais vantagens de um SGBD é que a mesma informação pode ser disponibilizada a utilizadores diferentes, ou seja, compartilhamento de dados.

Redundância minimizada: Os dados de um SGBD são mais concisos, porque, como regra geral, a informação nela aparece apenas uma vez. Isto reduz a redundância de dados, ou em outras palavras, a necessidade de repetir os mesmos dados uma e outra vez. Minimizando a redundância pode, portanto, reduzir significativamente o custo de armazenamento de informações em discos rígidos e outros dispositivos de armazenamento.

Precisão: dados precisos, consistentes são um sinal de integridade dos dados. SGBDs fomentam a integridade dos dados, porque as atualizações e alterações dos dados só tem que ser feitas em um só lugar. As chances de se cometer um erro são maiores se você é obrigado a alterar os mesmos dados em vários lugares diferentes do que se você só tem que fazer a mudança em um só lugar.

Utilização

5 NORMALIZAÇÃO

Para que não haja redundância de dados (duplicação de dados) ou valores inconsistentes na base de dados, algumas regras devem ser seguidas a fim de trabalhar-se com maior eficácia com os bancos de dados.

Para a criação de um banco de dados, devemos abstrair as entidades que compõem o sistema e a partir de cada uma, definir os atributos que cada um terá.

Estas definições nos levam a concretizar cada entidade em uma tabela e os atributos em seus campos.

Primeira Forma Normal: Uma relação se encontra na primeira forma normal se todos os domínios de atributos possuem apenas valores atômicos (simples e indivisíveis), e que os valores de cada atributo da entidade seja um valor simples. Assim sendo todos os atributos compostos devem ser divididos em atributos atômicos.

Segunda Forma Normal: Uma relação se encontra na segunda forma normal quando estiver na primeira forma normal e todos os atributos que não participam da chave primária são dependentes desta. Assim devemos verificar se todos os atributos são dependentes da chave primária e retirar-se da relação todos os atributos de um grupo não dependente que dará origem a uma nova relação, que conterá esse atributo como não a chave. Desta maneira, na segunda forma normal evita inconsistências devido a duplicidades.

Terceira Forma Normal: Uma relação estará na terceira forma normal, quando estiver na primeira forma normal e todos os atributos que não participam da chave primária são dependentes desta, porém não transitivos. Assim devemos verificar se existe um atributo que não depende diretamente da chave, retirá-lo criando uma nova relação que conterá esse grupo de atributos, e defina com a chave, os atributos dos quais esse grupo depende diretamente.(MARQUES, 2015)

6 DOMÍNIO (DOMAIN)

O sistema de classificação de candidatos do processo seletivo deverá ter um cadastro de pessoas. Estas pessoas serão de duas espécies: clientes e/ou funcionários.

Representa o conjunto de valores atômicos admissíveis de um componente (coluna) de uma relação (tabela)

- Telefone: conjunto de 10 dígitos.

- CPF: conjunto de 7 dígitos.

-Idade_Empregado: 16<=idade<=70.

Departamentos: conjunto de departamentos de uma empresa

A cada domínio está associado um tipo de dados ou formato.

Exemplo:

> Telefone(ddd) dddd-dddd em d = {0,1,2,…,9}

> IdadeEmpregado: número inteiro entre 16 e 70.

Um esquema de uma tabela R, definida por R(A1,A2,…,An), é um conjunto de atributos do tipo R = {A1,A2,…,An}.

Cada atributo Ai é o nome de um papel realizado por algum domínio D na tabelaR, ou seja, o nome de uma coluna na tabela R.

Restrições de domínios são estabelecidas determinando-se domínios de valores para cada coluna de uma tabela. Normalmente são estabelecidos e definidos os valores que uma coluna de uma tabela pode ter, isto é, o domínio da coluna.

Permitem, por exemplo:

> Verificar os valores inseridos em um banco de dados.

> Testar consultas para garantir que as comparações tenham sentido.

> Em geral o domínio é especificado por tipos primitivos de dados, tais como integer, float, char, date, time, money etc.

Também podem ser descritos pela definição de subconjuntos de tipos primitivos ou de listas enumeradas, ou seja, lista de valores possíveis de existir na coluna. (RODRIGUES, 2011)

Utilização

7 TRIGGERS

Um trigger é uma instrução que o sistema executa automaticamente como um efeito colateral de uma modificação no banco de dados. Para criar um mecanismo de trigger, temos de cumprir dois requisitos:

1. Especificar quando um trigger deve ser executado. Isso é desmembrando em um evento que faz com que o trigger seja verificado e uma condição que precisa ser satisfeita para que a execução do trigger prossiga

2. Especificar as ações a serem tomadas quando o trigger for executado.Esse modelo de trigger é conhecido como modelo evento-condição-ação para triggers.

O banco de dados armazena triggers como se fossem dados normais, de modo que sejam persistentes e acessíveis a todas as operações do banco de dados. Quando entramos com um trigger no banco de dados , o sistema de banco de dados assume a responsabilidade por executá-lo sempre que o evento especificado ocorre e a condição correspondente é satisfeita.

Utilização: A trigger se encontra na tabela isenção, a mesma ocorre depois de um update na tabela, quando a tabela update tiver o valor do atributo “Homologada” diferente de 0, significa que será verdade a aplicação da isenção da taxa, para tal o valor booleano da variável “Pago” da tabela taxa_inscrição, passa a ser 1.

8 PROCEDURE

Stored Procedure, que traduzido significa Procedimento Armazenado, é uma conjunto de comandos em SQL que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

Um Stored Procedure pode reduzir o tráfego na rede, melhorar a performance de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc.

Por todas estas e outras funcionalidades é que os stored procedures são de extrema importância para os DBAs e desenvolvedores.

Há cinco tipos de procedures básicos que podemos criar:

>Procedimentos Locais - São criados a partir de um banco de dados do próprio usuário;

>Procedimentos Temporários – Existem dois tipos de procedimentos temporários: Locais, que devem começar com # e Globais, que devem começar com ##;

>Procedimentos de Sistema – Armazenados no banco de dados padrão do SQL Server (Master), podemos identificá-los com as siglas sp, que se origina de stored procedure. Tais procedures executam as tarefas administrativas e podem ser executadas a partir de qualquer banco de dados.

>Procedimentos Remotos - Podemos usar Queries Distribuídas para tais procedures. São utilizadas apenas para compatibilidade.

>Procedimentos Estendidos - Diferente dos procedimentos já citados, este tipo de procedimento recebe a extensão .dll e são executadas fora do SGBD SQL Server. São identificadas com o prefixo xp.(CASERTA, 2017)

Utilização: foi criado uma procedure dentro do nosso banco de dados com a intenção de validar os dados. A mesma denomina-se procedure_pessoa(Insert_Atomic), e contém dentro dela uma transaction que “commita” os dados se tudo ocorreu de acordo com o esperado e da um Rollback se o contrário ocorrer.

9 INDEX

Índices nos bancos de dados são utilizados para facilitar a busca de informações em uma tabela com o menor número possível de operações de leituras, tornado assim a busca mais rápida e eficiente.

O exemplo clássico para explicar a utilização de índices é comparar uma tabela do banco de dados a uma lista telefônica, onde a mesma possui um índice por ordem alfabética do sobrenome dos “participantes”. Sabendo a letra inicial do

sobrenome é possível refinar a pesquisa iniciando a mesma pela página correspondente a letra do sobrenome.

O SQL Server utiliza o mesmo principio da lista telefônica gravando as informações dos índices em uma estrutura chamada de B-Tree.

Campos para serem indexados a fim de ganhar desempenho:

>Chaves Primárias;

>Chaves Estrangeiras;

>Colunas acessadas por ranges (between);

>Campos utilizados em group by ou order by;

Campos que não devem ser indexados:

>Campos dos tipos: text, image, decimais;

>Campos calculados;

>Campos com alta cardinalidade (Masculino ou Feminino);

(LOPES, 2017)

Utilização: Foi criada uma index na tabela pessoa, para facilitar a busca dos dados. Por default as chaves primárias já são uma forma de index, portanto utilizamos a data de nascimento para fazê-lo e com seu tipo B-TREE.

10 COMMIT E ROLLBACK

Se uma transação falhar por um motivo qualquer depois de atualizar o banco de dados, mas antes que a transação seja confirmada, pode ser preciso reverter(rollback) a transação. Se quaisquer valores de item de dados tiverem sido alterados pela transação e gravados no banco de dados, eles precisam ser restaurados para seus valores anteriores. As entradas de log do tipo undo são usadas para restaurar os valores antigo dos itens de dados que precisam ser revertidos.

Se uma transação T for revertida, qualquer transação S que tenha, enquanto isso, lido o valor de algum item de dados X gravado por T também deve ser revertida. De modo semelhante, quando S for revertida, qualquer transação R que tenha lido o valor de algum item de dados Y gravado por S também precisa

ser revertida, e assim por diante. Esse fenômeno é chamado de rollback-em-cascata (propagação de cancelamento), e pode ocorrer quando o protocolo de recuperação garante schedules recuperáveis, mas não garante schedules estritos ou sem propagação. É fácil entender que o rollback em cascata pode ser muito complexo e demorado, É por isso que quase todos os mecanismos de recuperação são projetados de modo que o rollback em cascata nunca seja necessário.(ELMASRI, 2011)

Utilização: o commit foi utilizado para realizar um insert dentro da procedure em uma transaction, sua função em específico é validar o cadastro e passar por parâmetros os dados, inserindo nas tabelas em questão, caso não seja possível em algum momento passar todos os dados devido a algum erro foi utilizado o Rollback. Tudo o que foi especificado até então se trata do conteúdo do Insert_Atomic.

11 JOIN

O comando JOIN do SQL tem a função básica de agregar tabelas mediante um campo que faça sentido às mesmas.

INNER JOIN

Utilizando o INNER JOIN você terá como resultado de sua consulta somente os pares de cliente/compra, ou seja, os casos em que uma compra não está ligada a nenhum cliente, ou cliente que não possuem compras, não serão apresentados no resultado.

LEFT JOIN

O LEFT JOIN traz todos os resultados da tabela mais à esquerda e o agrega ao seu valor correspondente das outras tabelas, caso existam. Ou seja, nesse caso a consulta retornaria todos os clientes e suas compras, caso existam. Caso não existam compras para essa cliente os campos relativos à tabela de compras ficariam em branco.

RIGHT JOIN

De forma análoga ao LEFT JOIN, o RIGHT JOIN traz todos os resultados da tabela mais à direita e o agrega ao seu valor correspondente das outras tabelas, caso existam. Ou seja, nesse caso a consulta retornaria todas as

compras e, caso existam, seus clientes. Caso não existam clientes para essa compra os campos relativos à tabela de clientes ficariam em branco.

OUTER JOIN

O comando OUTER JOIN, mesmo não sendo muito conhecido, pode ser bastante útil em alguns casos. Ele traria como resultado todos os clientes que não estão ligados à nenhuma compra.(RIBEIRO, 2016)

Utilização: os Joins foram utilizados dentro da view pessoa(view_pessoa_naturalidade_nacionalidade), o mesmo tem a função de comparar através de um **inner join** entre pessoaXcidade, cidadeXestado e estadoXpais, se os dados a serem obtidos estão contidos nas tabelas e conferem com as chaves a serem comparadas.

12 VIEW

A view pode ser definida como uma tabela virtual composta por linhas e colunas de dados vindos de tabelas relacionadas em uma query (um agrupamento de SELECT's, por exemplo). As linhas e colunas da view são geradas dinamicamente no momento em que é feita uma referência a ela.

Como já dito, a query que determina uma view pode vir de uma ou mais tabelas, ou até mesmo de outras views.

Observação: podemos realizar qualquer query por meio de views, assim como alterar dados por meio delas, o que é feito com algumas restrições.

Ao criarmos uma view, podemos filtrar o conteúdo de uma tabela a ser exibida, já que a função da view é exatamente essa: filtrar tabelas, servindo para agrupá-las, protegendo certas colunas e simplificando o código de programação.

É importante salientar que, mesmo após o servidor do SQL Server ser desligado, a view continua "viva" no sistema, assim como as tabelas que criamos normalmente. As views não ocupam espaço no banco de dados.

Vantagens das Views

Temos muitos motivos e vantagens para usarmos views em nossos projetos. A seguir são citados três que podem fazer a diferença:

>Reuso: as views são objetos de caráter permanente. Pensando pelo lado produtivo isso é excelente, já que elas podem ser lidas por vários usuários simultaneamente.

>Segurança: as views permitem que ocultemos determinadas colunas de uma tabela. Para isso, basta criarmos uma view com as colunas que acharmos necessário que sejam exibidas e as disponibilizarmos para o usuário.

>Simplificação do código: as views nos permitem criar um código de programação muito mais limpo, na medida em que podem conter um SELECT complexo. Assim, criar views para os programadores a fim de poupá-los do trabalho de criar SELECT's é uma forma de aumentar a produtividade da equipe de desenvolvimento.(BALBO, 2017)

Utilização: a view foi utilizada na tabela para **trazer os dados** das tabelas pessoa, cidade e pais(country), sua função neste BD é trazer o nome da pessoa, a naturalidade, bem como sua nacionalidade.

DESCRIÇÃO INFORMAL DOS DADOS

O sistema de classificação de candidatos do processo seletivo deverá ter um cadastro de pessoas. Estas pessoas serão de duas espécies: clientes e/ou funcionários.

Lembrando que uma ou várias pessoas possuem um ou mais telefones. E um ou mais telefones podem pertencer a uma ou mais pessoas.

Uma ou mais pessoas possuem um estado civil. E um estado civil pode pertencer a uma ou várias pessoas.

As pessoas acima citadas também podem ter um endereço. Onde uma pessoa pode ter apenas um endereço. E um endereço pode pertencer a uma ou várias pessoas.

Estes endereços contêm uma cidade. E um ou vários endereços podem pertencer a apenas uma cidade. Enquanto que uma cidade pode ter vários endereços.

Por sinal uma cidade está em algum estado. Para tal, uma ou mais cidades podem pertencer a apenas um estado. E um estado pode ter várias cidades.

Um determinado funcionário possui uma data de admissão e pertence a uma equipe. Estas (as equipes) são identificadas pelo código da equipe e diferenciadas uma da outra pelo nome e setor. Além do mais, um funcionário pode ter apenas um cargo ou vários. Já um cliente pode possuir um tipo, que demonstra se o mesmo é um cliente classic, silver, ou gold.

A empresa aérea vende diversas passagens de voos. Os voos pertinentes se diferenciam dos demais apenas por possuir uma especificação enquanto tal, que seria o código (cd_voo) referente ao voo em questão.

As vendas de passagem ocorre da seguinte maneira: Para cada venda, estarão contidas na passagem as informações do voo, avião, da equipe de funcionários, e do cliente. Um ou mais clientes podem pertencer a um ou mais voos. Enquanto que um voo ou mais pode ter um ou vários clientes. Além do mais na compra da passagem será definido o método de pagamento(cartão/boleto).

De posse da passagem um cliente faz checkin, mas para cada passagem o checkin pode ser feito uma única vez. Para auxiliar o entendimento, uma descrição mais estruturada das entidades detectadas e relacionamentos levantados, que representam o minimundo estão representadas abaixo:

Processo Seletivo

PESSOAS

>pessoa (cdpessoa, nome, data_nasc, sexo, RG, CPF, passaporte, nome_pai, nome_mãe, ativo, cd_est_civil)

> Tabela principal, na qual serão definidas as informações mais importantes do candidato ao processo seletivo

Notificação

>notificação(cdNotificação, cd_tipo_notificação)

>tipo notificação(cdTipoNotificação, descrição)

>a tabela notificação é usada para notificar de diferentes formas para cada tipo de usuario.

Emissor RG

>Emissor RG(emissor_RG, cd_pessoa)

>determina o local de emissão do RG

>entidade fraca da tabela pessoa

Estado civil

>estado civil(cdEstadoCivil, descrição)

>contém o status de relacionamento da pessoa

Telefone

>telefone(cdTelefone, telefone)

>telefone(cdTipoTelefone, descrição, cd_pessoa, cd_telefone)

>Contém o número de telefone de acordo com o tipo de telefone selecionado.

Usuario

>usuario(cd_pessoa, usuario, senha, email, tipo, data_cad, data_last_update)

>Contém o nome de usuário usado, senha, email e contém as datas de cadastro e do último acesso e o tipo de usuário cadastrado.

Escolaridade

>escolaridade(cdEscolaridade, descrição)

>tipo escolaridade(cdTipoEscolaridade, descrição, cd_pessoa, cd_escolaridade)

>Determina o tipo de escolaridade do usuário

Deficiencia

>deficiencia(cdDeficiencia, tipo, observações, cd_tipo_deficiencia, cd_pessoa)

>tipo deficiencia(cdTipoDeficiencia, descrição)

>Observa se a pessoa tem algum tipo de deficiência, se sim, determina qual o grau da deficiência.

Cidade

>cidade(cdCidade, nome_cidade, cd_estado)

>endereço(cdEndereço, CEP, Rua, Numero, Bairro, cd_cidade)

>estado(cd_Estado, nome_Estado, siglaEstado, cd_country)

>country(cdCountry, nome_Country, siglaCountry, nacionalidade)

>tipo endereço(cdTipoEndereço, descrição, cd_pessoa, cd_endereço)

>Estas tabelas determinam o país de origem, o estado, a cidade e o endereço do usuário.

Processo seletivo

>comissão(cdComissão, descrição, cd_pessoa, cd_processo_seletivo)

>processo_seletivo(cdProcessoSeletivo, data_inicio, data_fim, cd_tipo_processo)

>processo seletivo cota(cdProcessoSeletivoCota, vagas, cd_processo_seletivo, cd_cotas)

>cotas(cdCota, descrição)

>inscrição cotas(cdInscriçãoCotas, cd_inscrição, cd_cota)

>o processo seletivo contém uma comissão que avaliará se o usuario necessitará de cota, se sim, será homologado o número de vagas e o tipo de cota escolhido.

Inscrição

>inscrição(cdInscrição, data_inscrição, primeira_opção, segunda_opção, cd_processo_seletivo, cd_pessoa)

>curso(cdCurso, nome_curso)

>isenção(cdIsenção, motivo, homologada,cd_incrição)

>determina as opçoes de curso disponiveis pelo processo e se o usuário estara apto para a isenção a taxa de incrição

Taxa de Inscrição

>taxa inscrição(cdTaxaInscrição, data_vencimento_boleto, Taxa_inscrição, pago, data_pagamento)

>tipo processo seletivo(cdTipoProcesso, descrição, cd_taxa_inscrição)

>Relaciona o tipo do processo seletivo selecionado pelo usuário, e determina a taxa de inscrição que será cobrada, armazenando a data de pagamento e a data de vencimento da taxa.

REFERENCIAS

RODRIGUES, Felipe Nery. Banco de Dados: Projeto e Implementação. 2. ed. São Paulo: Érica, 2011. 400 p.(TRIGGER)

CASERTA, Thiago. **Introdução aos Stored Procedures no SQL Server.** Disponível em: <http://www.devmedia.com.br/introducao-aos-stored-procedures-no-sql-server/7904>. Acesso em: 13 set. 2017. (PROCEDURE)

LOPES, Nicholas. **Índices no SQL Server.** Disponível em: <http://www.devmedia.com.br/indices-no-sql-server/18353>. Acesso em: 13 set. 2017. (INDEX)

ELMASRI, Ramiz. **Sistemas de Banco de dados.** 6. ed. São Paulo: Pearson, 2011. 780 p. (ROLLBACK)

BALBO, Wellington. **Conceitos e criação de views no SQL Server.** Disponível em: <http://www.devmedia.com.br/conceitos-e-criacao-de-views-no-sql-server/22390>. Acesso em: 13 set. 2017. (VIEWS)

RIBEIRO, Gabriella Fonseca. **SQL – CONSULTAS COM JOIN.** 2016. Disponível em: <http://eufacoprogramas.com/sql-consultas-com-join/>. Acesso em: 13 set. 2017. (JOINS)

MARQUES, Diego. **Normalização de Dados SQL.** 2015. Disponível em: <https://www.portaleducacao.com.br/conteudo/artigos/informatica/normalizacao-de-dados-sql/60735>. Acesso em: 13 set. 2017. (NORMALIZAÇÃO)