



Sistemas de Informação 2

2015/16 Inverno

Trabalho Prático - fase 1

Grupo 19

Henrique Calhó nº38245

Tiago Magalhães nº28593

Docente

Nuno Datia

Índice

[Introdução](#)

[Modelo EA](#)

[Código SQL](#)

[Conhecimentos adquiridos](#)

Introdução

O trabalho tem como propósito criar a base de dados de um sistema de tickets para ser usado em balcões de apoio ao cliente. Para além disso, tem de ser criado todas as ferramentas que permitem manipular os dados.

Neste relatório pretendemos apresentar a nossa implementação e as explicações necessárias das soluções adoptadas.

Como conclusão serão analisados os conhecimentos adquiridos.

Modelo EA

Em anexo é enviado o modelo EA.

Os nomes de todas as entidades e atributos foram escolhidos com cuidado para que pudessem ser perceptíveis. De qualquer modo, há certos detalhes que necessitam de explicação.

- **Request**

- representa um pedido de informação sobre um ticket feito por um utilizador
- o pedido e a resposta são ambos representados nesta entidade, sendo o technician associado referente ao atributo response onde fica guardada a resposta ao pedido

- **Step**

- representa um passo necessário à resolução de um tipo de ticket

- **Ticket_Action**

- representa acções que podem ser efectuadas sobre o ticket para a sua resolução
- é necessário garantir que o passo associado à acção pertence ao mesmo tipo que a acção

- **Ticket_Log**

- regista alterações feitas sobre um ticket
- qualquer instrução que desencadeie alterações na tabela ticket fica aqui registada

Todas as especificações do enunciado que não são garantidas pelo modelo ea são implementadas nos procedimentos armazenados que manipulam os dados.

Código SQL

No script creates em anexo está presente todo o código que resolve as alíneas do enunciado. Para as últimas alíneas é também entrega uma pasta de testes com um script para cada. Os procedimentos armazenados criados estão também comentados com a pergunta a que correspondem.

Para as alíneas mais complexas em que a solução não seja imediata vamos apresentar uma explicação do que fazemos e como cumprimos os propositos do enunciado.

d) Inserir, remover e actualizar informação de um ticket

“Note que, além disso, não pode ser removido do sistema as informações relativas a

tickets e acções a eles associadas.”

Para garantir isto, adicionámos à tabela ticket um atributo visible. Assim os tickets em vez de serem removidos são apenas escondidos do utilizador.

Criámos também as views vi_Ticket, vi_Ticket_Log, vi_Request, vi_Ticket_Action para assim só apresentar ao utilizador tickets e informação respectiva dos tickets válida.

sp_Assign_Technician_To_Tickets

g) Actualizar o prioridade de um ticket

A prioridade de um ticket é manipulada pelo procedimentos sp_Update_Ticket_Priority, sp_Increate_Ticket_Priority e pela função func_Increase_Priority.

“Note que a cada 7 dias, o ticket sobe de prioridade, de forma automática.”

Para garantir esta automatismo pensámos que a melhor solução seria um job que invocaria sp_Increate_Ticket_Priority.

A função é auxiliar. Serve para obter o próximo nível de prioridade.

Se o técnico responsável quiser actualizar a prioridade utiliza-se sp_Update_Ticket_Priority.

h) Atribuir um técnico responsável a todos os tickets no estado 'Waiting'

A atribuição do ticket é feita automaticamente quando ele é inserido através do procedimento `sp_Insert_Ticket`.

Criamos a função `func_Select_Technician_With_Least_Tickets` que obtém o técnico que tem menos tickets atribuídos no momento, e o procedimento `sp_Insert_Ticket` trata de o inserir no ticket.

Caso por alguma razão um ticket se encontre sem técnico a procedimento `sp_Assign_Technician_To_Tickets` trata disso.

k) Inserir de forma automática o registo de alterações efectuadas sobre os tickets

Os registos de alterações são guardados na tabela `Ticket_log`.

Para registar alterações sobre os tickets criamos o trigger `tr_On_Ticket_Changes` que obtém a instrução sql que desencadeou a execução do trigger e de seguida chama o procedimento `sp_Insert_Ticket_Log`.

l) Inserir uma resposta a um pedido de informação

Os pedidos de informação de e resposta correspondente são ambos guardados na tabela `Request`.

Os pedidos são feitos pelo utilizador através do procedimento `sp_Request_Ticket_Info` e os técnicos respondem através do procedimento `sp_Answer_Ticket_Request`.

O tempo de resposta a cada pedido pode ser obtido pela chamada a `func_Calculate_Response_Time`.



m) Obter os n tickets mais prioritários.

A função `func_Get_Most_Priority_Tickets` retorna uma tabela ordenada com os tickets que se querem.

n) e o) Listar os tickets. Obter a informação de um ticket, usando o seu identificador.

Para estas duas alíneas entendemos que o objetivo é apenas apresentar a informação pretendida. Portanto criámos procedimentos que o fazem.

Se na segunda fase esta informação for necessária para trabalhar, em vez de apenas listar a informação, precisaremos de criar tabelas temporárias para as utilizar.

p) Remover um ticket.

Está incluída na alínea d.

Conhecimentos adquiridos

Como conclusão principal, o grupo pensa ter atingido o objetivo principal que acompanhava a criação deste sistema, nomeadamente:

1. Desenvolver um modelo de dados adequado aos requisitos, normalizado até à 3NF
2. Desenvolver uma solução baseada em bases de dados dinâmica, adequada aos requisitos
3. Utilizar correctamente vistas

Devemos usar views se pretendermos manter estabilidade entre versões, garantir que código antigo continua a funcionar mesmo com alterações do esquema ou restringir acesso a informação.

Nós utilizámo-las para esconder a visibilidade dos “removidos” e a informação relativa.

4. Utilizar correctamente procedimentos armazenados

Os nossos procedimentos são responsáveis por todas as alterações sobre a base de dados.

5. Utilizar correctamente gatilhos

Criámos um trigger para registar alterações sobre uma tabela e assim poder criar um log.

6. Utilizar correctamente funções

Definimos algumas funções para calcular e obter informação auxiliar a ser utilizada pelos procedimentos.

7. Utilizar correctamente níveis de isolamento
8. Utilizar correctamente controlo transaccional