Regras de negócio

Para garantir que uma pessoa não pode ser, simultaneamente, paciente e médico da mesma consulta foi realizada uma função chamada *MedicoNotPaciente*, que verifica com um *Inner Join* da tabela *Medico* com a tabela *Paciente* através dos bi’s que o médico e o paciente não são a mesma pessoa. Foi também realizada uma função chamada *FaturasEmDivida* para verificar se um paciente tem ou não mais de 5 faturas não pagas, o que leva à impossibilidade de marcar uma consulta.

Uma vez que a posologia de um medicamento só podia ser selecionado de uma certa gama de valores, foi criada uma tabela com o nome *Posologia*, que apenas permite armazenar os valores das mesmas. Foi tomada esta decisão sobre a de usar um *check*, visto que existia a possibilidade de adicionar/alterar/remover alguns dos valores das posologias, sendo assim mais fácil a alteração dos mesmos via *insert/update/delete* em vez de um *alter table*.

A regra do cálculo do montante da fatura foi resolvida através da realização de 3 *triggers after insert/update/delete*, ou seja depois de cada operação sobre um *ItemFatura* é realizado um *update* na tabela *Fatura* que atualiza o valor do montante para o correto utilizando as tabelas *inserted* e *deleted* correspondentemente. Foi criada a função *VerificaPessoa* para garantir que os dados da pessoa numa fatura têm os mesmos valores na tabela *Pessoa*, isto é, a função é usada quando é se tenta passar uma fatura, para verificar que todos os dados estão corretos.

Para o motivo de uma consulta, foi criada uma nova tabela com os valores que podem ser selecionados. A decisão foi tomada, mais uma vez, para facilitar a adição/edição/remoção dos valores. O preço da consulta a ser colocado em fatura tem como valor base 40€, isto foi realizado através do *DEFAULT* utilizado na criação da tabela.

Durante a realização do ponto 10, foi adicionado um campo chamado *temDesconto*, que funciona como uma *flag*, que indica se o paciente terá desconto na próxima consulta, ou seja, quando é emitida uma consulta vai ser verificado o se o paciente terá direito a bónus, se tiver, será realizado do respetivo desconto e serão então adicionados os pontos correspondentes, e a *flag* é colocada a *“false”*. Caso o paciente não tenha direito a desconto, são apenas somados os pontos e se caso disso colocada a *flag* a *“true”* para a próxima consulta.

Para o cumprimento do ponto 11 foi necessário a inserção de novas tabelas para se guardar toda a informação necessária sobre o Paciente que se pretende eliminar, o respetivo EA encontra se em anexo. Caso o paciente que se pretende eliminar tiver dívidas este não pode ser eliminado (o estado das faturas é emitida). Caso isso não se verifique é copiada toda a informação do Paciente, presente nas tabelas Pessoa, Paciente, MedicamentoPaciente, Consulta, Morada e Telefone para as respetivas tabelas de histórico. Só depois de essa cópia ser efetuada é que se procede a eliminação do Paciente. Contudo, como um médico pode ser simultaneamente um paciente e ambos derivam de Pessoa é importante que neste caso nem a informação de Pessoa, Morada e Telefone sejam eliminados. Assim sendo elimina-se a informação das tabelas MedicamentoPaciente, Consulta e Paciente e só depois de se verificar se o Paciente não um Medico, através da função VerificarMedico, é que se elimina a informação em Morada, Telefone e Pessoa. Este procedimento armazenado tem um nível transacional elevado, Serializable, isto para evitar o possível aparecimento de Phantom, para que não seja possível por exemplo o aparecimento de uma nova morada, ou número de telefone durante a execução do procedimento.