MAC 426 — Sistemas de Banco de Dados

Segundo Semestre de 1997[1]

Descrição inicial do projeto

Este documento visa fornecer uma idéia geral do projeto a ser desenvolvido, especificando com que tipo de software se pode trabalhar neste projeto. Eu aproveito para descrever coisas que não fazem parte do nosso cronograma, daí o enunciado do projeto ficar um tanto longo :-).

1 Introdução

O objetivo deste trabalho é expor o aluno (você!) aos tipos de questões envolvidas no trabalho na área de *banco de dados*. Uma questão importante é **o que é um banco de dados**. Obter uma definição ou resposta precisa para o termo causa muito debate entre os profissionais de banco de dados. Dependendo da experiência da pessoa e o que se está tentando fazer, um banco de dados pode ser construído de maneira a ser muitas coisas. Em classe, esboçamos uma definição formal restrita à categoria de *bancos de dados relacionais*.

Num sentido mais amplo, um banco de dados pode ser considerado como a soma total de todas as informações presentes na organização a que o banco de dados se refere. Esta definição genérica incluíria os dados armazenados em coisas definidas tradicionalmente como banco de dados, e mais os dados armazenados em arquivos e fitas espalhados por toda a organização, e até os dados impressos ou datilografados em formulários, circulares e outras fontes "não computacionais". Embora esta definição faça sentido em muitos cenários, é genérica demais para o que estamos tentando estudar neste semestre. Adotaremos então uma definição mais restrita:

> Um *banco de dados* é uma coleção de dados organizados logicamente e gerenciados por um conjunto unificador de princípios, procedimentos e funcionalidades que ajudam a garantir a consistência da aplicação e interpretação dos dados na organização".

Sob o ponto de vista do nosso projeto, eu espero que vocês:

[1] Agradecimentos ao Marcelo, que me forneceu estas notas, e à Dilma, que foi quem as editou pela primeira vez.

- modelem a organização (sob um dado ponto de vista parcialmente restrito ao tipo de ação desejada em cima do conjunto de dados, ou seja, não é possível dentro do espectro de um curso de um semestre com quatro créditos modelar a organização como um todo e nem prever que tipo de evolução a organização pode prever no momento, mas é possível que vocês definam uma parte da organização de forma coerente no projeto) usando como ferramenta o modelo entidade-relacionamento;

- derivem um modelo relacional e validem este modelo, em relação ao modelo entidade-relacionamento desenvolvido no item anterior, com as ferramentas estudadas em classe;

- especifiquem o conjunto de consultas a serem oferecidas para o usuário em termos de funcionalidade e interface a ser oferencia para o usuário;

- programem um conjunto representativo das consultas planejadas;

- analisem o desempenho do seu banco de dados sob o ponto de vista de quantidade de informação que pode ser armazenada e o tempo de execução das consultas;

- apontem os problemas enxergados no seu banco de dados.

Além disso, vocês devem se familiarizar com um produto de banco de dados[2]. Faz parte do projeto o trabalho de familiarização com o produto sendo utilizado. O último ítem é muito importante, pois saber realizá-lo significa "ter experiência na área de bancos de dados".

2 Produtos

Este projeto deve ser implementado em sistemas gerenciadores de banco de dados que se encaixem em uma das duas arquiteturas de banco de dados apresentadas aqui. A arquitetura de um banco de dados descreve a organização lógica e operacional básica que um banco de dados em particular assumirá na execução das tarefas desejadas. É a arquitetura do produto que define como o banco de dados será visto pelos usuários e programadores.

[2]Um produto de banco de dados é um software (programa ou coleção de programas) que gerencia o armazenamento e recuperação de um conjunto de dados, organiza-o logicamente e provém o usuário com certas funcionalidades que garantem que os dados estarão logicamente organizados de forma consistente.

2.1 Produtos relacionais

Existem *muitos* produtos que podem ser vistos como relacionais, nos quais a manipulação de dados é no estilo SQL. Entre eles destacamos: DB2, SQL/DS, SQL/400 (todos da IBM), Allbase (HP), Interbase, Informix, Oracle, Sybase, Access, Ingres e muitos outros. Se vocês têm acesso a algum desses softwares, sejam benvindos a usá-lo como instrumento de implementação deste projeto. O MS Access está disponível nos micros do CEC. O *ZIM*, creio, também está disponível nos micros do CEC, e é uma alternativa excelente para quem procura um bom gerenciador de banco de dados (e de fácil aprendizado). O DB2 estará disponível no CEC dentro em breve.

Não é possível classificar com certeza um dado produto como sendo relacional ou não. Mas é inegável que o enfoque relacional foi muito importante na indústria pela criação do SQL que se estabeleceu num período relativamente curto como um *standard* para linguagens de banco de dados. Observe que é do interesse de um vendedor aderir a uma linguagem de acesso standard, já que ela garante que o perfil da linguagem fique consistente com os dos outros vendedores. Isto também significa que usuários podem mudar de um produto para outro, já que quando você sabe SQL você sabe como usar vários bancos de dados igualmente bem. E já que a maioria dos bancos de dados fala a mesma língua, eles podem falar um com o outro. Devemos notar que em terms de SQL e standards para bancos relacionais nenhum fabricante adere aos standards cem por cento, muitos fabricantes aderem consideravelmente, e muitos produtos se encaixam no esquema pelo menos um pouco.

Mas não esqueçamos de olhar para estes padrões propostos (no caso o modelo relacional com uma linguagem de consulta tão standard e uma estrutura conceitual tão desenvolvida) tentando notar os problemas. A linguagem SQL, que estudaremos, é acessível ao usuário "não perito" o suficiente para que este consiga especificar muitas consultas; mas este aspecto "user friendly" não implica que o banco de dados é "amigável": num banco de dados relacional o usuário pode construir consultas que sejam muito ineficientes. Dar ao usuário poderes para acessar os dados de qualquer forma que ele queira pode implicar que o acesso é feito de maneira incorreta. **Perigo:** os usuários podem tentar fazer o banco de dados fazer coisa demais. Embora um banco de dados relacional possa gerenciar relacionamentos complexos entre os dados, tentar fazer isto com o sistema de forma indisciplinada pode resultar em tempos de respostas do sistema deploráveis, e programas muito complicados.

2.2 Produtos não relacionais, voltados para PCs

Na seção anterior enumeramos muitos produtos importantes do mercado, a maior parte deles usados para gerenciar base de dados de grande dimensão, e disponíveis em máquinas de grande e médio porte[3]. Mas existem produtos que ainda tomam uma fatia grande do mercado. Eles podem não fornecer tudo que você espera de um banco de dados, mas eles são usados em todos os lugares. Eles são produtos desenvolvidos para o mercado de computadores pessoais, que continua crescendo (especialmente agora com as redes locais interconectando PCs e conectando-os a máquinas de médio porte que funcianam como *servidores* (prestadores de serviço) principais. Os representantes desta categoria de produtos que considero aceitáveis para exploração do problema sendo analisado neste curso são: dBase III, dBase IV e FoxBase[4].

O enfoque destes produtos para gerenciamento de dados permite o programador/usuário trabalhar com vários arquivos ao mesmo tempo. Navegação entre os arquivos (no nosso caso, estes arquivos representarão relações) é alcançada ao se estabelecer um relacionamento lógico entre os arquivos e pelo movimento de um *apontador para o record corrente*. É possível estabelecer índices que dizem ao sistema que campos do arquivo serão usados pelas buscas primárias e operações de navegação.

Do nosso ponto de vista, tais produtos são "primitivos", pois não estão fundamentados nem por uma teoria que garanta sua corretudo, nem por métodos de desenvolvimento que garantam a practicidade de seu uso. Se você vai usar um dos produtos desta linha, tenha em mente que é esperado que você mantenha a disciplina e o encadeamento lógico do uso de um banco de dados relacional. Você terá que prover uma justificativa convincente para utilizá-los. Hoje em dia, por meio de interfaces padrões, já é possível acessar tais bancos de dados por uma forma restrita de SQL. Por exemplo, o ODBC (Open Database Conectivity) é um padrão lançado no mercado pela Microsoft para acessos a Bancos de Dados em ambientes Microsoft. Você provavelmente necessitará de um driver ODBC para o seu banco de dados específico (relacional ou não) se quiser acessar o banco de dados através de uma linguagem de programação "externa" ao produto do banco de dados (e.g. C, C++, etc), ou seja, que não faz parte do ambiente de desenvolvimento eventualmente provido junto com o produto. Se você for usar um produto não relacional, recomenda-se fortemente o uso de tal interface.

[3] Alguns deles também estão disponíveis no mundo DOS.

[4] Aceito discussões envolvendo o *Paradox*, mas a justificação tem que ser convincente.

3 Que problema modelar? (ou: banco de dados de que?)

Como já foi dito muitas vezes, não há restrição em termos do domínio[5] a ser escolhido para este projeto. Eu fornecerei aqui algumas sugestões, mas nenhuma delas será especificada com detalhe. O ideal seria cada grupo encontrar um problema que ofereça alguma motivação. Se alguém do grupo tem uma motivação profissional na linha de desenvolver produtos e ir à luta para tentar vendê-los, esta é uma oportunidade para exercitar a capacidade de trabalho *free-lancer*[6], isto é, de escolher um área não muito explorada e conseguir produzir um protótipo capaz de incentivar potenciais compradores a investir no produto. Com o passar dos anos a gente vai juntando anedotas de pessoas que ganharam dinheiro desenvolvendo bancos de dados para gerenciar desde linhagem de gado em fazendas até manufatura de bijouterias. Sua criatividade e disposição para definir um problema "tentador" do ponto de vista de mercado conta pontos positivos.

Mas claramente nem todos tem tendência profissional para este tipo de desenvolvimento de sistemas. Assim, você pode seguir uma das seguintes sugestões:

1. Silva é o gerente de uma pequena loja de calçados localizada em um grande shopping center. Ele anda tendo muitos problemas em lidar com o inventário da loja. No passado um gerente precisava manter contabilidade de 40 ou 50 estilos de sapatos apenas, mas agora o inventário foi quadruplicado para poder enfrentar a concorrência.

2. Coelho possui uma loja de animais. Ele gostaria de manter um cadastro dos animais que foram vendidos e para quem, assim como um histórico de cada animal atualmente em venda. Deveria ser possível, por exemplo, automaticamente gerar um cartão de aniversário para os animais já vendidos, na tentativa de se manter na visão do cliente. A loja também fornece o serviço de "cabelereiro" e tosagem, logo um sistema simples para marcar horas seria interessante.

[5] Este é o nome que se dá em engenharia de software para o "problema" sendo tratado, isto é, para a parte do mundo real sendo considerado no *design*.

[6] Por aí existem muitos empreendedores que por algum motivo trabalham em projetos avulsos além do emprego fixo das nove às seis. Acredito que a estrutura em termos de tempo neste projeto pode ser semelhante ao que se encontra disponível quando se trabalha. Claro que aqui não há a motivação financeira e — o que realmente empurra o projeto para o fim — o stress de um compromisso assumido com um cliente.

3. Ferreira possui uma clínica de acompanhamento psicológico. A idéia seria manter as informações básicas sobre o cliente (quem indicou, endereço, telefone, etc). Existem várias formas de terapia (individual, em grupo, para casais, etc) e os psicólogos que trabalham na clínica são especializados em determinadas categorias. Nesta clínica que o paciente *não* tem um profissional designado a ele de forma constante: para cada consulta a ser marcada pode ser colocado qualquer profissional com experiência no tipo de terapia sendo usado no paciente.

4. Santos vem há muito tempo colecionando discos e livros. Ele resolveu que chegou a hora de cadastrar o seu acervo e informatizar o seu empréstimo de livros a amigos, inclusive usando o conhecimento das áreas de interesse dos seus amigos para sugerir títulos que venham a interessá-los. Ele gostaria de obter listas com base em autores e estilos também.

5. Gama é proprietário de uma mecânica de automóveis. Ele gostaria de usar um computador para guardar informações como clientes, seus carros e seus problemas, assim como funcionários e fornecedores de peças.

6. Menezes é dono de restaurantes, e no atual contexto procura maneiras de se tornar *fashionable*, conseguindo portanto projeção na imprensa e destaque no concorrido mundo gastrônomico paulistano. Ele pensou em seguir o esquema de muitos restaurantes americanos onde a interação entre o garçom e a cozinha se dá através de um computador. O sistema deveria gerenciar informações como pratos disponíveis, preços, ingredientes e pedidos.

Todas estas sugestões representam aplicações bem pequenas, que obviamente podem se tornar complicadas conforme se vai adicionando detalhes.

4 Primeira Etapa: Modelo Descritivo (data de entrega: terça-feira, 23/09/97)

Cada grupo deve entregar um relatório com a especificação do domínio a ser explorado, e em que nível de detalhe. É a chamada *análise de requisitos* de seu sistema. Enumere o tipo de informação a ser armazenado, quais as manipulações que serão feitas. A idéia é que este relatório reflita o resultado de sua conversa com seu cliente. Ao ler este relatório eu devo ser capaz de

entender qual o problema que vocês atacarão e que tipo de uso dos dados a serem armazenados vocês têm em mente. Prove que vocês *pensaram* no projeto e no objetivo a ser atingido pelo software.

Importante: jamais encare tal documento como algo que serve para o professor apenas. Ele será a base de desenvolvimento do seu sistema. Sua grande utilidade está:

- principalmente na sua preparação, que o força a pensar sobre o assunto. Logo, clareza e consistência são aspectos essenciais.
- em constituir parte fundamental da memória de seu projeto. Ela comunicará ao desenvovedores das fases futuras (neste caso: você, amanhã) os objetivos do projeto. E, obviamente, necessitara de correções e revisões durante o processo de desenvolvimento.

Mãos a obra e boa sorte.