Universidade Federal da Fronteira Sul
Notas de Aula sobre Modelo Relacional – Estudo de Caso
Banco de Dados I
Prof. Denio Duarte

O texto abaixo apresenta um estudo de caso para exemplificar como os dados podem ser armazenados no formato relacional.
Este exemplo modela um sistema simples de controle de consultas de uma clínica médica.
A clínica precisa dos dados dos pacientes, dos médicos e das consultas agendadas.

Abaixo, um exemplo de relatórios do sistema:

PACIENTES

Nome	Data Nascimento	Cidade	UF	CPF
João da Silva	20/08/1975	Chapecó	SC	512 512 111 10
Alan Turing	23/06/1912	Londres	UK	111 111 222 10
Edgar Codd	23/08/1923	Port. Island	UK	234 123 123 33

MÉDICOS

Nome	Data Nascimento	Cidade	UF	Especialidade
Gregory House	15/06/1964	Chicago	MI	Clínico Geral
Chris Taub	12/02/1970	New York	NY	Clínico Geral
Carlos Chagas	09/07/1879	Rio	RJ	Infectologista

CONSULTAS

Médico	Paciente	Data	Hora
Gregory House	Alan Turing	10/05/2013	14:00
Carlos Chagas	Edgar Codd	11/05/2013	09:00
Gregory House	Edgar Codd	11/05/2013	15:00
Carlos Chagas	João da Silva	12/05/2013	09:30

Imagine como os dados desses relatórios estariam arrmazenados em um BD relacional. Quantas tabelas seriam necessárias para armazenar esses dados?
Pelo menos 3 tabelas: armazenar os dados dos médicos, paciente e consultas.

Relembrando a notação vista na última aula presencial:

```
nometabela(chave, normal, opcional, chave estrangeira (origem))
```

Começamos com a tabela **Paciente**. Baseado nos dados do relatório acima, vemos que existe o atributo CPF que é o nosso candidato a chave primária (vamos chamar a partir de agora a chave primária de PK). Assim, a primeira tentativa de modelar a tabela seria:

```
paciente(cpf, nome, dtnasc, ncid, uf)
```

Com a instância (baseada no relatório acima):

cpf	**nome**	**dtnasc**	**cidade**	**uf**
51251211110	João da Silva	20/08/1975	Chapecó	SC
11111122210	Alan Turing	23/06/1912	Londres	UK
23412312333	Edgar Codd	23/08/1923	Port. Island	UK

Passamos para a tabela **Medico**. Observando o relatório, temos as dicas dos atrbutos mas nenhum que possa ser a PK. Primeira tentativa

`medico(nome, dtnasc, ncid, uf, espec)`

Quem seria a PK? Poderíamos incluir um atributo para armazenar o CPF e transformar ele em PK. Assim, a tabela ficaria:

`medico(cpf, nome, dtnasc, ncid, uf, espec)`

Porém, vamos imaginar que o CPF não é interessante para o cadastro desta aplicação e que a aplicação precisa de atributos característicos de médicos. Uma nova opção seria o número do conselho regional de medicina (CRM) que é único para cada médico. Então, vamos substituir o CPF pelo CRM. Agora a versão final da tabela seria:

`medico(crm, nome, dtnasc, ncid, uf, espec)`

Perceba que o relatório acima com a relação dos médicos não apresenta o CRM. Isso mostra um exemplo de visão externa dos dados (arquitetura 3 camadas vista em aula).
A instância seria:

crm	**nome**	**dtnasc**	**ncid**	**uf**	**espec**
332	Gregory House	15/06/1964	Chicago	MI	Clínico Geral
145	Chris Taub	12/02/1970	New York	NY	Clínico Geral
966	Carlos Chagas	09/07/1879	Rio	RJ	Infectologista

Finalmente, o maior desafio: a tabela que armazena as consultas. A primeira tentativa seria fazê-la exatamente como o relatório das consultas mostra e ficaria assim:

`consulta(nomeme, nomepa, datac, horac)`

A instância seria:

nomeme	**nomepa**	**datac**	**horac**
Gregory House	Alan Turing	10/05/2013	14:00
Carlos Chagas	Edgar Codd	11/05/2013	09:00
Gregory House	Edgar Codd	11/05/2013	15:00
Carlos Chagas	João da Silva	12/05/2013	09:30

Essa solução não condiz com o modelo relacional pois o modelo prega que as tabelas que utilizam dados de outras tabelas (utilizando chave estrangeira – FK) devem ter um “ponteiro” para a tabela de origem. Por que isso? Perceba que os nomes dos médicos e dos pacientes já se encontram nas tabelas Medico e Paciente, respectivamente. Assim, basta ter este ponteiro para as respectivas tabelas para encontrar o nome (leia o material *Lecture Notes on Relational Model* no Moodle).
Este ponteiro aponta para a PK da tabela de origem. Assim, vamos subsituir os nomes pelas PK das tabelas envolvidas. Nova tentativa:

`consulta(crm(medico), cpf(paciente), datac, horac)`

A instância seria:

crm	cpf	datac	horac
332	11111122210	10/05/2013	14:00
996	23412312333	11/05/2013	09:00
332	23412312333	11/05/2013	15:00
996	51251211110	12/05/2013	09:30

Se quisermos saber o nome do médico de uma determinada consulta, basta utilizar o **crm** da tabela **Consulta** e pesquisar na tabela **Medico**.
Próximo passo: descobrir a PK.
Tentativa 1:

`consulta(crm(medico), cpf(paciente), datac,horac)`

Problema: se o **crm** é a PK quer dizer que apenas um valor de CRM pode aparecer na tabela, em outras palavras, um médico só poderia fazer uma consulta na clínica pois se fosse inserida uma nova consulta para o médico, o SGBD retornaria um erro de "duplicate PK", ou seja, chave primária duplicada. Na nossa instância acima, as tuplas com 332 não poderiam ter duas ocorrências. O mesmo vale para as tuplas com 996.
A primeira solução que vem a cabeça seria fazer uma PK composta com **crm** e **cpf**.

`consulta(crm(medico), cpf(paciente), datac, horac)`

Observando a instância da tabela consulta, parece que a nova PK atende às necessidades da aplicação. Porém, vamos supor a seguinte situação, o João da Silva marcou uma consulta com Carlos Chagas para mostrar os exames no dia 20/05/2013 às 17:00, ou seja, teríamos que incluir na tabela a tupla

`<996,51251211110,20/05/2013,17:00>`

Se tentarmos inserir essa tupla, o SGBD retornará "duplicate key" pois a chave <996 ,51251211110> já existe na tabela.
Solução? Podemos acrescentar a data da consulta como chave:

`consulta(crm(medico), cpf(paciente), datac, horac)`

Aparentemente, isso resolveria o nosso problema. A nova instância ficaria:

crm	cpf	datac	horac
332	11111122210	10/05/2013	14:00
996	23412312333	11/05/2013	09:00
332	23412312333	11/05/2013	15:00
996	51251211110	12/05/2013	09:30
996	51251211110	20/05/2013	17:00

Perceba que agora a PK é composta por 3 atributos: crm, cpr e datac. Assim, não existe repetição de valores com a combinação desses atributos.
Essa solução ainda traz um problema: um mesmo médico não pode atender o mesmo paciente no mesmo dia pois o SGBD não deixaria incluir uma tupla com o mesmo crm, cpf e datac.
A solução 100% segura seria:

`consulta(crm(medico), cpf(paciente), datac, horac)`

Ou seja, todos os atributos da tabela compõe a chave.
Alguns desenvolvedores não apreciam esse tipo de solução e criam chaves primárias "artificiais" para resolverem o problema do uso de chave composta. Por exemplo, a tabela Consulta poderia ser criada como:

`consulta(codc, crm(medico), cpf(paciente), datac, horac)`

Agora a PK é o atributo **codc** (código da consulta).
Vantagens? A referência a PK é menor, basta utilizar o **codc**.

Desvantagens? A verdadeira chave não será controlada pelo SGBD, ou seja, deve-se criar meios para garantir que não existem ocorrências iguais de consulta na tabela. Por exemplo, duas secretarias diferentes da clínica inserem a mesma consulta no sistema.
Aparentemente temos o banco de dados pronto.
Porém podemos fazer mais uns ajustes para ele atender bem o que o modelo relacional preconiza (indica).
Na tabela Medico existe um atributo que possui um valor textual (sequência de caracteres) que armazena valores repetidos: a especialização do médico (`espec`). Se ao invés de uma clínica fosse um grande hospital com centenas de médicos, o atributo especialização possuiria vários valores repetidos na tabela. Por exemplo, se existissem 30 médicos ortopedistas, o valor `ortopedista` apareceria em todas as tuplas de tais médicos.
Solução? Criaríamos uma tabela para armazenar as especialidades existentes e transformariam o atributos `espec` da tabela Medico para uma FK.
Vamos lá. Primeiro a nova tabela

```
especialidade(codesp, descr)
```

Com a seguinte instância:

codesp	**descr**
1	Clinico Geral
2	Infectologista
3	Otorrinolaringolista
4	Ortopedista

Temos que modificar a tabela Medico.

```
medico(crm, nome, dtnasc, ncid, uf, code(especialidade))
```

A instância ficaria assim:

crm	**nome**	**dtnasc**	**ncid**	**uf**	**code**
332	Gregory House	15/06/1964	Chicago	MI	1
145	Chris Taub	12/02/1970	New York	NY	1
966	Carlos Chagas	09/07/1879	Rio	RJ	2

Legal, não? :)

EXERCÍCIO (entregar via Moodle conforme data colocada no sistema)

Dado o seguinte relatório de uma locadora de veículos

placa	modelo	ano	cliente	dt locação	dt devolução	promo	diária
MK-10	Audi Q3	2019	C11-Edgar Codd	01/08/2020	10/08/2020	blue	80,00
KK-10	Ka	2018	C22-Alan Turing	25/09/2020		green	40,00
XX-77	HR-V	2020	C34-Tanenbaum	15/08/2020	23/08/2020	blue	80,00
MK-10	Audi Q3	2019	C41-Marie Curie	20/08/2020	24/08/2020	green	40,00
KK-10	Ka	2018	C05-Grace Hopper	01/07/2020	30/07/2020	red	100,00
XX-77	HR-V	2020	C34-Tanenbaum	17/09/2020	21/09/2020	red	100,00

Baseado no estudo de caso apresentado, crie os esquemas das tabelas de uma banco de dados relacional que atenda o armazenamento dos dados do relatório. Não esqueça de apresentar também as instâncias das tabelas. Você pode criar atributos que julgue importante nas tabelas. Por exemplo, a tabela Carro pode ter a cor, montadora. Já a tabela cliente pode ter o email, endereço, etc.