

Um modelo relacional de dados para Grandes bancos de dados compartilhados

1. Modelo Relacional e Forma Normal

A visão relacional (ou modelo) dos dados descritos em Seção 1 parece ser superior em vários aspectos ao gráfico ou modelo de rede [3, 4] atualmente em voga para sistemas inferenciais. Ele fornece um meio de descrever dados com sua estrutura natural apenas - isto é, sem qualquer estrutura adicional para representação da máquina. Consequentemente, fornece uma base para um alto nível linguagem de dados que irá render a máxima independência entre os programas, por um lado, e a representação da máquina e organização de dados, por outro.

Um modelo relacional de dados é proposto como uma base para proteger os usuários de sistemas de dados formatados mudanças potencialmente perturbadoras na representação de dados causados pelo crescimento do banco de dados e mudanças no tráfego.

Uma outra vantagem da visão relacional é que forma uma base sólida para o tratamento de derivabilidade, redundância, e consistência das relações - estes são discutidos na Seção 2 .

Um modelo baseado em relações n-árias, uma normal formulário para relações de banco de dados, e o conceito de um universal sublinguagem de dados são introduzidos. Na Seção 2, certas operações ções sobre as relações (além da inferência lógica) são discutidas e aplicado aos problemas de redundância e consistência no modelo do usuário.

É proposto que a maioria dos usuários deve interagir com um modelo relacional dos dados consistindo em uma coleção de relações que variam no tempo (ao invés de relações). Cada o usuário não precisa saber mais sobre qualquer relacionamento do que seu nome junto com os nomes de seus domínios. Mesmo essas informações podem ser oferecido em estilo de menu pelo sistema (sujeito a segurança e restrições de privacidade) mediante solicitação do usuário.

Normalmente, existem muitas maneiras alternativas em que um remodelo nacional pode ser estabelecido para um banco de dados. Em a fim de discutir uma forma preferida (ou forma normal), nós deve primeiro introduzir alguns conceitos adicionais (ativos domínio, chave primária, chave estrangeira, domínio não simples) e estabelecer alguns links com a terminologia atualmente em uso na programação de sistemas de informação. No restante de neste artigo, não devemos nos preocupar em distinguir entre relações e relações, exceto onde parece vantagens tageful para ser explícito.

Considere um exemplo de um banco de dados que inclui relações relativas a peças, projetos e fornecedores. Um relação chamada *parte* é definida nos seguintes domínios:

- (1) número da peça
- (2) nome da peça
- (3) cor da parte
- (4) peso parcial
- (5) quantidade disponível
- (6) quantidade no pedido e possivelmente outros domínios também. Cada um desses domínios

é, com efeito, um conjunto de valores, alguns dos quais ou todos podem ser representado no banco de dados a qualquer instante.

Em contraste, os problemas tratados aqui são os de *dados independência* - a independência dos programas de aplicação e atividades terminais de crescimento em tipos de dados e mudanças na representação de dados - e certos tipos de *dados inconsistência* que se espera que se torne problemática mesmo em sistemas não-redutores.

1.4. FORMA NORMAL

Uma relação cujos domínios são todos simples pode ser representada enviado em armazenamento por um homo-coluna bidimensional matriz genética do tipo discutido acima. Um pouco mais estrutura de dados complicada é necessária para uma relação com um ou mais domínios não simples.

Na Seção 2, operações sobre relações e dois tipos de redundância são definidas e aplicadas ao problema de manter os dados em um estado consistente. Isso está vinculado a tornar-se um sério problema prático à medida que mais e mais diferentes tipos de dados são integrados em comum bancos de dados.