Modelo Relacional de Base de Dados

Os futuros usuários de grandes bancos de dados devem ser protegidos contra ter que saber como os dados são organizados na máquina (o representação interna). Um serviço de alerta que fornece essas informações não são uma solução satisfatória. Atividades de usuários nos terminais e a maioria dos programas de aplicativos deve permanecer não afetado quando a representação interna dos dados é alterada e mesmo quando alguns aspectos da representação externa são alterados. Mudanças na representação de dados serão frequentemente necessária como resultado de mudanças na consulta, atualização e relatório tráfego e crescimento natural dos tipos de informação armazenada. Os sistemas de dados formatados não inferenciais existentes fornecem aos usuários com arquivos estruturados em árvore ou rede um pouco mais geral modelos dos dados. Na Seção 1.

Os sistemas de dados formatados não inferenciais existentes fornecem aos usuários com arquivos estruturados em árvore ou rede um pouco mais geral modelos dos dados. Na Seção 1, inadequações desses modelos são discutidos. 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.

PALAVRAS-CHAVE E FRASES: **banco de dados, banco de dados, estrutura de dados, dados organização, hierarquias de dados, redes de dados, relações, derivabilidade, redundância, consistência, composição, junção, linguagem de recuperação, predicado cálculo, segurança, integridade de dados.**

1. Modelo Relacional e Forma Normal

1.1. INTRODUÇÃO

Este artigo está preocupado com a aplicação de elementos teoria da relação mentária para sistemas que fornecem acesso a grandes bancos de dados formatados. Exceto para um papel por Childs [1], a principal aplicação das relações com dados sistemas tem sido os sistemas de resposta a perguntas dedutivas. Levein e Maron [2] fornecem inúmeras referências para trabalhar nesta área.

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.

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 superim- apresentando qualquer estrutura adicional para representação da máquina finalidades. Consequentemente, fornece uma base para um alto nível linguagem de dados que irá render a máxima independência ser-.

2. O modelo de rede, por outro lado, gerou um número de confusões, a menos das quais é o engano a derivação de conexões para a derivação de rela- (consulte as observações na Seção 2 sobre ″ armadilha de *conexão* "). Finalmente, a visão relacional permite uma avaliação mais clara do escopo e limitações lógicas do presente formatado sistemas de dados, e também os méritos relativos (de uma lógica ponto de vista) de representações concorrentes de dados dentro de um sistema único. Exemplos dessa perspectiva mais clara são citado em várias partes deste artigo. Implementações de sistemas para apoiar o modelo relacional não são discutidos.

Ampla, a coleção de relações consiste apenas no seguinte relação de formação: *funcionário (número de série, nome, gerente #, managername)* com *serial #* como chave primária e *gerenciador #* como estrangeiro chave. Vamos denotar o domínio ativo por, ~, e supor que P (s #, d #, ...), Q (s #, j #, .-.), R (d #, j #, ...), *Em (no gerente) c A, (no de série) Em (managername) C At (name)* e para sempre t. Neste caso, a redundância é óbvia: o *managername do* domínio é desnecessário. Para ver que é um redundância forte, conforme definido acima, observamos que *~ 34 (funcionário) = ~ '12 (funcionário) l [lm (funcionário).* No segundo exemplo, a coleção de relações inclui um relação S descrevendo fornecedores com chave primária s #, uma relação D que descreve departamentos com chave primária d #, a relação J que descreve projetos com chave primária j #, e o seguintes relações: onde em cada caso .-. denota domínios diferentes de s #, d #, j #. Vamos supor que a seguinte condição C é conhecida por manter independente

do tempo: departamento de suprimentos do fornecedor d (relação P) se e somente se o fornecedor fornecer algum projeto j (relação Q) à qual d é atribuído (relação R). Então nós pode escrever a equação ~ **'12 (P) = 71-12 (Q).** 71 "21 **(R)** e, portanto, exibem uma grande redundância. Uma razão importante para a existência de forte re- Dundâncias no conjunto nomeado de relacionamentos é o usuário con- venience. Um caso particular disso é a retenção de semi- relacionamentos obsoletos no conjunto nomeado de modo que os antigos pro- gramas que se referem a eles pelo nome podem continuar a funcionar corretamente retamente. Conhecimento da existência de fortes redundâncias no conjunto nomeado permite que um sistema ou banco de dados administre maior liberdade na seleção do representante armazenado para lidar de forma mais eficiente com o tráfego atual. Se o redundâncias fortes no conjunto nomeado são refletidas diretamente em redundâncias fortes no conjunto armazenado (ou se outro redundâncias são introduzidas no conjunto armazenado), então, gen- de modo geral, o espaço de armazenamento extra e o tempo de atualização são consumido com uma possível queda no tempo de consulta para alguns consultas e em carga nas unidades centrais de processamento.

2.2.2. *Redundância fraca.* Um segundo tipo de redun- dança pode existir. Em contraste com a redundância forte, não é caracterizado por uma equação. Uma coleção de relações é *fracamente redundante* se contiver uma relação que tenha um projeto ção que não é derivada de outros membros, mas está em todas as vezes uma projeção de *alguma* junção de outras projeções de relações na coleção. Podemos exibir uma redundância fraca tomando o segundo exemplo (citado acima) para uma redundância forte, e como- supondo agora que a condição C não é válida o tempo todo.

2.4. RESUMO

Na Seção 1, um modelo relacional de dados é proposto como um base para proteger os usuários de sistemas de dados formatados de as mudanças potencialmente perturbadoras na representação de dados causados pelo crescimento do banco de dados e mudanças no tráfego. Uma forma normal para a coleção variável no tempo de relação navios é introduzido.