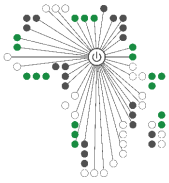


Universidade Virtual Africana

INFORMÁTICA APLICADA: CSI 4306

SISTEMAS AVANÇADOS DE BASE DE DADOS

Aurélio Armando Pires Ribeiro

Prefácio

A Universidade Virtual Africana (AVU) orgulha-se de participar do aumento do acesso à educação nos países africanos através da produção de materiais de aprendizagem de qualidade. Também estamos orgulhosos de contribuir com o conhecimento global, pois nossos Recursos Educacionais Abertos são acessados principalmente de fora do continente africano.

Este módulo foi desenvolvido como parte de um diploma e programa de graduação em Ciências da Computação Aplicada, em colaboração com 18 instituições parceiras africanas de 16 países. Um total de 156 módulos foram desenvolvidos ou traduzidos para garantir disponibilidade em inglês, francês e português. Esses módulos também foram disponibilizados como recursos de educação aberta (OER) em oer.avu.org.

Em nome da Universidade Virtual Africana e nosso patrono, nossas instituições parceiras, o Banco Africano de Desenvolvimento, convido você a usar este módulo em sua instituição, para sua própria educação, compartilhá-lo o mais amplamente possível e participar ativamente da AVU Comunidades de prática de seu interesse. Estamos empenhados em estar na linha de frente do desenvolvimento e compartilhamento de recursos educacionais abertos.

A Universidade Virtual Africana (UVA) é uma Organização Pan-Africana Intergovernamental criada por carta com o mandato de aumentar significativamente o acesso a educação e treinamento superior de qualidade através do uso inovador de tecnologias de comunicação de informação. Uma Carta, que estabelece a UVA como Organização Intergovernamental, foi assinada até agora por dezenove (19) Governos Africanos - Quênia, Senegal, Mauritânia, Mali, Costa do Marfim, Tanzânia, Moçambique, República Democrática do Congo, Benin, Gana, República da Guiné, Burkina Faso, Níger, Sudão do Sul, Sudão, Gâmbia, Guiné-Bissau, Etiópia e Cabo Verde.

As seguintes instituições participaram do Programa de Informática Aplicada: (1) Université d'Abomey Calavi em Benin; (2) Université de Ougagadougou em Burkina Faso; (3) Université Lumière de Bujumbura no Burundi; (4) Universidade de Douala nos Camarões; (5) Universidade de Nouakchott na Mauritânia; (6) Université Gaston Berger no Senegal; (7) Universidade das Ciências, Técnicas e Tecnologias de Bamako no Mali (8) Instituto de Administração e Administração Pública do Gana; (9) Universidade de Ciência e Tecnologia Kwame Nkrumah em Gana; (10) Universidade Kenyatta no Quênia; (11) Universidade Egerton no Quênia; (12) Universidade de Addis Abeba na Etiópia (13) Universidade do Ruanda; (14) Universidade de Dar es Salaam na Tanzânia; (15) Universite Abdou Moumouni de Niamey no Níger; (16) Université Cheikh Anta Diop no Senegal; (17) Universidade Pedagógica em Moçambique; E (18) A Universidade da Gâmbia na Gâmbia.

Bakary Diallo

O Reitor

Universidade Virtual Africana

Créditos de Produção

Autor

Aurelio Ribeiro

Par revisor(a)

Martina Barros

UVA - Coordenação Académica

Dr. Marilena Cabral

Coordenador Geral Programa de Informática Aplicada

Prof Tim Mwololo Waema

Coordenador do módulo

Robert Oboko

Designers Instrucionais

Elizabeth Mbasu

Benta Ochola

Diana Tuel

Equipa Multimédia

Sidney McGregor

Barry Savala

Edwin Kiprono

Kelvin Muriithi

Victor Oluoch Otieno

Michal Abigael Koyier

Mercy Tabi Ojwang

Josiah Mutsogu

Kefa Murimi

Gerisson Mulongo

Direitos de Autor

Apoiado por

Projeto Multinacional II da UVA financiado pelo Banco Africano de Desenvolvimento.

Tabela de conteúdo

Descrição Geral do Curso

Bem-vindo(a) ao módulo de Sistema Avançado de Base de Dados

Neste módulo, vamos estudar os fundamentos necessários para compreender o Sistema Avançado de Base De Dados. Além disso, o conteúdo deste módulo lhe ajudará a ter uma visão geral sobre complexidade da base de dados. O módulo é uma continuidade do primeiro módulo (Principios de Sistemas de Base de Dados) e aborda conceitos mais avançados.

Para cada unidade, o módulo fornece as actividades de ensino e de aprendizagem por forma a levar o(a) estudante a aprender os temas propostos neste módulo.

As actividades podem ser feitas em grupo ou individualmente e você (s) podem consultar outras bibliografias por forma a complementar os seus estudos na area de base de dados. Finalmente, segue uma avaliação sumativa para ajudar-lhe a avaliar a qualidade da sua aprendizagem.

Pré-requisitos

Para estudar este módulo você deverá ter conhecimentos prévios básicos de Principios de Sistemas de Base de Dados.

Materiais

Os materiais necessários para completar este curso incluem:

- Laboratório de informática
- Softwares (Mysql_Server 5.x, Oracle 11g,)
- Ligação à Internet

Objetivos do módulo

O módulo é uma continuidade do estudo iniciado em Principios de Sistemas de Base de Dados. Você irá aprender a criar, implementar e gerir uma Base de dados, entretanto, após terminar este modulo, espera-se que você seja capaz de:

- Explicar a importância das tecnologias de base de dados e gestão de dados;
- Desenvolver uma base de dados com base no paradigma Orientado a Objectos;
- Propor estratégias para lidar com grandes volumes de dados;
- Gerir os dados do mercado;
- Explicar as diferentes estratégias de transação em bases de dados;
- ·Explicar e implementar a integração da segurança e recuperação em sistemas de base de dados;
- Implementar um sistema de armazenamento simples.

Unidades

- Unidade 0: Diagnóstico
- Unidade 1: Conceitos de Base de Dados Orientada a Objectos
- Unidade 2: Processamento de consultas e Transações de Base de Dados
- Unidade 3: Data Warehouse e data Mining
- Unidade 4: Segurança de base de dados

Avaliação

Em cada unidade encontram-se incluídos instrumentos de avaliação formativa a fim de verificar o progresso do(a)s estudantes.

No final de cada módulo são apresentados instrumentos de avaliação sumativa, tais como testes e trabalhos finais, que compreendem os conhecimentos construidos e as competências desenvolvidas no módulo.

A implementação dos instrumentos de avaliação sumativa fica ao critério da instituição que oferece o curso.

Calendarização

Unidade	Temas e Atividades	Estimativa do tempo
1.Diagnostico	1.Conceitos de princípios de Sistema de base de dados	02h
1.Conceitos de Base de Dados Orientada a Objectos	1.1. Base de dados Orientada a Objecto 1.2. Caraterísticas da Base de dados Orientada a Objecto 1.3. Linguagem e desenho de Base de Dados	18h
2.Processamento de consultas e Transações de Base de Dados	2.1. Processamento de consultas 2.2. Transacções de Base de Dados 2.3. Operações em Transacções de Base de Dados	20h
3.Data Warehouse e data Mining	3.1. Base de Dados Distribuidas 3.2. Conceitos Data Warehouse 3.3. Conceitos Data Mining	30h
4.Segurança de base de dados	4.1. Tipos de Segurança 4.2. Controlo de Acesso à Base de Dados 4.3. Backup 4.4. Recovery	40h

Leituras e outros Recursos

As leituras e outros recursos deste curso são:

Unidade 1

Leituras e outros recursos obrigatórios:

- Tecnologia de Bases de Dados (3ª Edição), José Luis Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431
- Introdução a Sistemas de Base de Dados, DATE, C. J.. Elsevier Editora, 2004.

Unidade 2

Leituras e outros recursos obrigatórios:

- Tecnologia de Bases de Dados (3ª Edição), José Luis Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431

Unidade 3

Leituras e outros recursos obrigatórios:

- Tecnologia de Bases de Dados (3ª Edição), José Luis Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431

Unidade 4

Leituras e outros recursos obrigatórios:

- Tecnologia de Bases de Dados (3ª Edição), José Luis Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431.

Unidade 0. Diagnóstico

Introdução à Unidade

O propósito desta unidade é verificar os seus conhecimentos relacionados com base de dados. Como estudante/formando/aluno, você vai encontrar questões de auto-avaliação que irá ajudá-lo a testar a sua prontidão para estudar este módulo. Você tem que encarar esta unidade como uma base para poder triunfar no estudo deste módulo.

Como instrutor, as questões de pré-avaliação fornecidos aqui levam os estudante/formando a saber se eles estão preparados, no entanto, Sugere-se que você incentive os estudantes a auto-avalição, respondendo a todas as perguntas abaixo. Caso os estudantes/formandos compreendam os pré-requisitos eles têm a base necessária para iniciar o Módulo.

Objectivos da Unidade

Após a conclusão desta unidade, deverás ser capaz de:

1. Estabelecer uma relação da interdisciplinaridade;
2. Analisar o seu grau de conhecimentos e competências para começar este módulo.

Esta avaliação visa fazer uma pré-avaliação para sua introdução a este módulo. Preste atenção ao resolver.

Escolha a opção que serve para responder correctamente à questão:

1. Quando é preferível usar um Sistema de Ficheiros?
   A. Sempre
   B. Nunca se deve usar
   C. Quando a Base de Dados e aplicações forem muito simples, bem definidas e não se espera que mude muito.
   D. Quando a base de dados estiver online.
   E. Nenhuma das anteriores

2. Para que serve uma Base de Dados

   A. Para aumentar a rentabilidade

   B. Para aumentar a velocidade do computador

   C. Para gerir grandes conjuntos de informação.

   D. Para eliminar redundância

   E. Nenhuma das anteriores

3. Qual é Comando para criar tabelas numa base de dados?

   A. DELETE.

   B. CREATE

   C. ALTER

   D. DO

   E. Nenhuma das anteriores

4. Qual deles não é um SGBD

   A. Oracle

   B. Excel

   C. Mysql

   D. Interbase

   E. Postgres

5. Em base de dados uma relação equivale a:

   A. Campo

   B. Linha

   C. Tabela

   D. Nenhuma das anteriores

6. é a associação de um ou mais atributos que em conjunto identificam cada um dos tuplos (linhas da tabela)

   A. Chave candidata

   B. Superchave

   C. Chave primaria

   D. Chave estrangeira

7. Estabeleça a correspondência

A.	Select		
B.	Create	1.	DCL
C.	Drop		
D.	Revoke		
E.	Insert	2.	DDL
F.	Grant		
G.	Update		
H.	Alter	3.	DML
I.	Delete		

8. Que solução deve ser adoptada no modelo relacional para relacionamentos M:N?

_____

_____

_____

_____

9. Explique quando uma tabela está em conformidade com cada uma das seguintes Formas Normais:

_____

_____

_____

_____

10. Calcule R x S [B = C]

R

A	B
a1	b1
a2	b2

S

C	D
b2	d2
b1	d3

11. Seja dada a tabela : PESSOA (Id, Nome, Idade, Salário, Telefone, Cod_Postal, Localidade)

a. Mostre apenas o nome e o telefone das pessoas que são de Marracuene cujo salário esta abaixo de 1000

Álgebra Relacional:

SQL:

b. Quais são as pessoas que não são de Mocuba?

Álgebra Relacional:

SQL:

Correção da Avaliação

1. Solução (C)
2. Solução (C)
3. Solução (B)
4. Solução (B)
5. Solução (C)
6. Solução (B)
7. Solução:

   DML (SELECT, INSERT, UPDATE, DELETE, …),

   DDL (CREATE, ALTER, DROP, …),

   DCL (GRANT, REVOKE, …)

8. Solução: No modelo relacional não é possível efectuar este tipo de relacionamento de forma directa. Neste caso, deve-se construir uma terceira tabela (tabela de associação). Essa tabela deve possuir chave primária composta de dois campos e as chaves estrangeiras provenientes das duas tabelas originais. Concluindo, um relacionamento de muitos-para-muitos deve ser dividido em dois relacionamentos de um-para-muitos com uma terceira tabela.
9. Solução: -- Uma tabela está na 1FN (Primeira Forma Normal) quando não possui tabelas aninhadas

- Uma tabela está na 2FN quando, além de estar na Primeira Forma Normal, não contém dependências parciais
- Uma tabela está na 3FN quando, além de estar na 2FN (Segunda Forma Normal), não contém dependências transitivas.

10. Solução:

R X S [B = C]

A	B	C	D
a1	b1	b1	d3
a2	b2	b2	d2

11. Solução - Alínea (A)

Álgebra Relacional:	$\pi_{Nome,Telefone}(\sigma_{Localidade='Marracuene' \wedge salario<1000}^{(PESSOA)})$
SQL:	SELECT Nome, Telefone FROM PESSOA WHERE Localidade =' Marracuene' AND salario <1000;

Solução - Alínea (B)

Álgebra Relacional:	$\pi_{Nome}(\sigma_{Localidade \neq 'Marracuene'}^{(PESSOA)})$
SQL:	SELECT Nome FROM PESSOA WHERE Localidade < >' Marracuene';

Conclusão da Unidade 0

Esta unidade tinha em vista verificar as suas competências básicas para poder trabalhar neste módulo. Caso não tenha conseguido resolver a maior parte das questões, procure ler sobre os conteúdos abordados em forma de questões na avaliação acima exposta. Se conseguir responder as questões, então está apto para iniciar este módulo. Votos de bom estudo.

Unidade 1. Conceito de Base de Dados Orientada a Objectos

Introdução à Unidade

Nos últimos anos, a procura por ferramentas que facilitem a integração entre o mundo orientado a objectos e o mundo relacional é cada vez mais crescente. Ferramentas de mapeamento objecto -relacional (O-R) nada mais são do que um "tradutor" entre duas linguagens diferentes.

Como em todas as traduções, algumas subtilezas da língua acabam sendo perdidas ou utilizam-se muito mais palavras para expressar um conceito que é relativamente simples na língua de origem. No mapeamento O-R não é diferente, acaba-se perdendo uma série de recursos da programação OO, ou tem que se escrever muito mais código para simular na base de dados relacional, algo simples na linguagem.

Armazenar objectos em uma BD relacional é uma tarefa difícil pois este modelo não possui mecanismos necessários para representar características básicas do modelo OO. Todos os desenvolvedores das linguagens orientadas a objectos sabem das dificuldades de passar um modelo orientado a objectos para uma persistência relacional.

Grande parte do tempo de desenvolvimento de um software é perdida na fase de mapeamento. Utilizando uma BDOO é possível eliminar ferramentas e códigos para o mapeamento objecto-relacional e aproveitar os benefícios do paradigma orientado a objectos sem estar preso pela BD, permitindo modelos de objectos mais ricos.

Nesse cenário, uma base de dados relacional não traz nenhuma vantagem. Talvez a escolha de um modelo Orientado a Objecto seja uma evolução natural.

Objetivos da Unidade

Após a conclusão desta unidade, deverá ser capaz de:

1. Identificar as carateristicas de um modelo OO;
2. Listar os SGBDOO.
3. Descrever a evolução e tendências das bases de dados;
4. Listar os aspetos relacionados com sistemas estendidos.

Termos-chave

Base de dados Orientação a objectos., Linguagem Orientadas a Objectos, Base de Dados Objecto-Relacional, Base de Dados Estendido

Actividades de Aprendizagem

Actividade 1.1. Base de dados Orientada a Objecto

É preciso uma Introdução à actividade

Contextualização

O modelo de Base de Dados Orientado a Objectos é relativamente recente. Os investigadores começaram a interessar-se por este modelo de BD no final de 1970, princípios de 1980, à medida que os conceitos de objecto e de programação orientada por objectos ia se enraizando e desenvolvendo.

A necessidade de representar realidades complexas levou ao desenvolvimeto de sistemas orientados para objectos. O objectivo da existência deste tipo de base de dados é permitir estender o conceito de programação orientada para objectos e adiciona-lá também aos sistemas de armazenamento de dados. Daqui resulta uma proximidade muito maior entre a aplicação e os elementos que são armazenados.

A base de dados orientada a objecto é um modelo em que cada informação é armazenada na forma de objectos, e só pode ser manipulada através de métodos definidos pela classe em que esteja o objecto. O conceito de base de dados OO é o mesmo da LOO (linguagem orientada a objecto), havendo uma pequena diferença: a persistência de dados.

Conceitos de Base de Dados Orientada a Objecto

Identidade de Objecto

Segundo Elmasri e Navathe (2005), um sistema de base de dados OO fornece uma identidade única para cada objecto independente armazenado na BD. Essa identidade única é geralmente implementada através de um identificador de objecto , ou OID único, gerado pelo sistema. O valor de um OID não é visivel para o utilizador externo, mas é utilizado pelo sistema para identificar cada objecto univocamente e para criar e gerenciar referências inter-objecto .

A principal propriedade exigida por um OID é que ele seja imutável, isto é, o valor do OID para um objecto não deve se alterar, com isso preserva-se a identidade do objecto do mundo real, que está sendo representado.

Para se garantir a imutabilidade do OID, SGBDOO devem possuir algum mecanismo para gerar OIDs, pois é desejavel que cada OID seja utilizada apenas uma vez, ou seja, mesmo que um objecto seja removido da BD, o seu OID não deve ser atribuido a outro objecto . Essas duas propriedades implicam que o OID ao deve depender de qualquer valor de atributo do objecto e nem deve ser baseado no endereço físico do objecto armazenado uma vez que esses valores, atributo e endereço físico podem ser modificados ou corrigidos.

Estrutura do Objecto

Ainda segundo Elmasri e Navathe (2005), em BDOO, o estado, valor corrente de um objecto complexo, pode ser construido a partir de outros objectos ou outros valores, utilizando certos construtores de tipo. Os objectos podem ser visualizados como um trio (i,c,v), onde "i" é um identificador de objecto único (OID), "c" é o construtor de tipo (ou seja, indicação de como o estado do objecto é construido) e "v" é o estado do objecto (ou valor corrente).

Os construtores geralmente utilizados são atom (atômo), tupla, set (conjunto), mas outros construtores como list, bag e array também podem ser utilizados. Os construtores do tipo set (conjunto), list, bag e array s, os chamados de tipos de coleção para distingui-los dos tipos básicos e das tuplas.

A principal característica de um tipo coleção é que o estado do objecto será uma coleção de objecto s que podem ser não ordenados (como set ou bag) ou ordenados (como list ou array). O construtor do tipo tupla é geralmente chamado de tipo estruturado (struct type), uma vez que corresponde ao construtor de struct nas linguagens de programação C e C++.

O construtor de átomos é utilizado para representar todos os valores atômicos básicos, como números inteiros, números reais, strings de caracteres, booleanos e quaisquer outros tipos de dados que o sistema diretamente suporta (Elmasri e Navathe,2005).

O estado do objecto "v" de um objecto (i, c, v), É interpretado com base no construtor "c".

- Para c=atom, o estado (valor) "v" É um valor atômico do domínio de valores suportados pelo sistema.
- Para c=set, o estado "v" É um conjunto de identificadores de objecto {i1, i2, i3..., in} que são os OIDs para um conjunto de objecto s que são tipicamente do mesmo tipo.
- Para c=tupla, o estado "v" È uma tupla da forma <a1:i1, a2:i2 ... an:in> onde cada aj é um nome de atributo e cada ij é um OID.
- Para c=list(lista), o valor "v" é uma lista ordenada {i1, i2,..., in} de OIDs de objecto s do mesmo tipo e estes estão ordenados.

- Para c=array, o estado do objecto é uma disposição (array) unidimensional de identificadores de objecto . A principal diferença entre uma array e uma lista é que uma lista pode possuir um número arbitrário de elementos, enquanto que um array geralmente possui um tamanho máximo e a diferença entre set e o bag é que todos os elementos num conjunto devem ser diferentes, enquanto que em uma bag pode possuir elementos duplicados.

Objecto

Os objectos são abstrações de dados do mundo real, com uma interface de nomes de operações e um estado local que permanece oculto. As abstrações da representação e das operações são ambas suportadas no modelo de dados orientado a objectos, ou seja, são incorporadas as noções de estruturas de dados e de comportamento. Um objecto tem um estado interno descrito por atributos que podem apenas ser acessados ou modificados através de operações definidas pelo criador do objecto. Um objecto individual é chamado de instância ou ocorrência de objecto. A parte estrutural de um objecto (em banco de dados) é similar à noção de entidade no modelo Entidade-Relacionamento.

Objectos Complexos

Segundo Elmasri e Navathe (2005), a capacidade de lidar com objectos complexos foi um dos principais motivos para a criação do modelo BDOO, pois na época não havia nenhum sistema que oferecesse tal recurso.

Existem dois tipos de objectos complexos: os não-estruturados e os estruturados.

- Objecto complexo não-estruturado é um objecto que utiliza um tipo que não é padrão na BD e podem ocupar uma grande·área de armazenamento. Estes objectos são do tipo **BLOB**, que significa Binary Large Object, e podem armazenar, por exemplo, os dados referentes a uma imagem bitmap, um texto formatado, uma musica entre outros.

Elmasri e Navathe (2005), afirmam que, por padão, os SGBDOOs não interpretam os dados contidos nestes objecto s, fornecendo apenas a funcionalidade de recuperar as informações parciais ou totais deste objecto para a aplicação. No entanto, o SGBDOO permite a criação de um tipo abstrato para estes tipos de objectos, e permite implementar as operações necessárias para definir o comportamento deste objecto. Assim, conclui-se que um SGBDOO È um sistema de tipo extensivel, ou seja, pode-se criar bibliotecas de tipos de objecto s abstratos, e reutilizar em outras aplicações.

Para exemplificar, supondo-se que se deseja armazenar dados referentes as impressões digitais em um banco de dados de investigações policiais. É necessário utilizar um objecto complexo não-estruturado. Assim, implementam-se as operações, por exemplo, a que faz o processamento de como localizar uma digital dentre as armazenadas no banco.

- Objecto complexo estruturado difere do não-estruturado pelo fato do SGBDOO conhecer a estrutura interna deste objecto . Ele é constituido da aplicação dos diversos construtores, de forma repetida. Por exemplo, um objecto complexo da classe "Caixa Postal" pode conter os objecto s locais número_caixa_postal e logradouro, objecto s que referenciam a outros objecto s, como cidade e estado, e objecto s que s, o conjunto de objecto s, como bairros.

Polimorfismo

Segundo Larman (2000), as base de dados OO também estão preparadas para o

conceito de polimorfismo, ou sobrecarga de operador. Este conceito permite que um mesmo nome de operação, aplicado a objectos de classes diferentes e que tenham uma superclasse em comum, possa ter implementaçõoes diferentes. Quando uma mensagem é passada ao objecto , o SGBDOO seleciona a implementação adequada de acordo com o tipo do objecto.

Por exemplo: A partir da classe Veículo, considere uma nova classe Barco, como na Figura 1. Note que os comportamentos (métodos) "Move" e "Para" são comuns às três classes. Quando o programa estiver em execução, se o método "Move" for invocado na classe Veículo, as especificidades de cada uma das classes serão utilizadas a fim de causar o efeito desejado, independente de qual seja essa classe, mas pelo tipo da instância do objecto.

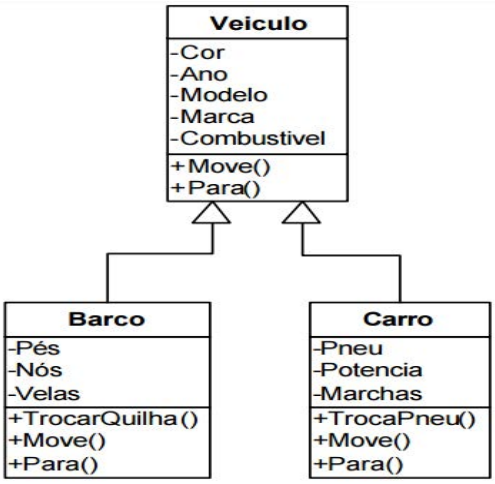


Figura 1 – Classes Carro e Barco especializadas em Veículo.

Encapsulamento

O conceito de encapsulamento está bastante ligado à abstração de dados, permitindo que apenas as informações que o objecto julgue apropriadas sejam visualizadas externamente, de acordo com o contexto da aplicação. Esse conceito também pode ser utilizado para acoplar dados com operações comumente executadas, como obter a idade de um Veículo. Ainda, se fosse o caso, seria possível tornar o atributo Ano invisível para os objectos externos e permitir a visualização apenas da operação Idade.

Modelos de classes

Ao utilizar uma BDOO, existe a possibilidade da utilização de um mesmo modelo de classe gerado para o desenvolvimento da aplicação na criação da BD, conforme visto na Figura 2. Geralmente, esse modelo é desenvolvido em UML (Unified Modeling Language).

Segundo Elmasri e Navathe (2005), o termo classe é geralmente utilizado para fazer referência a uma definição de tipo de objecto, juntamente com as definições das operações para aquele tipo. Um número de operações é declarado para cada classe de objecto e a assinatura (interface) de cada operação é incluida na definição da classe.

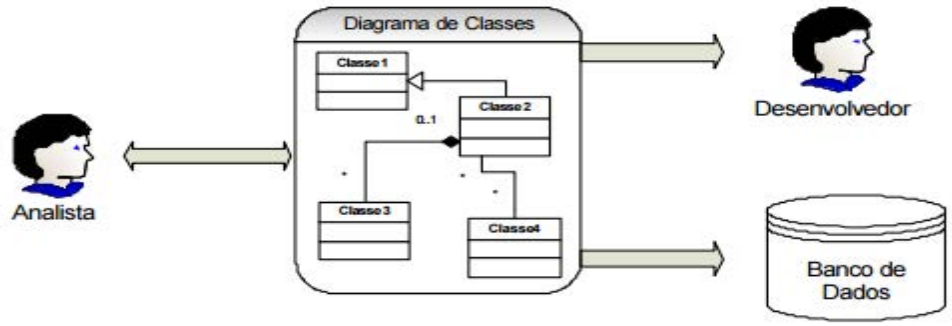


Figura 2 – Modelo de Classes .

Em geral, a declaração de uma classe é dividida em duas partes: as variáveis de instância (atributos de um objecto a ser instanciado) e a sua interface, que são operações (métodos) disponíveis para a classe.

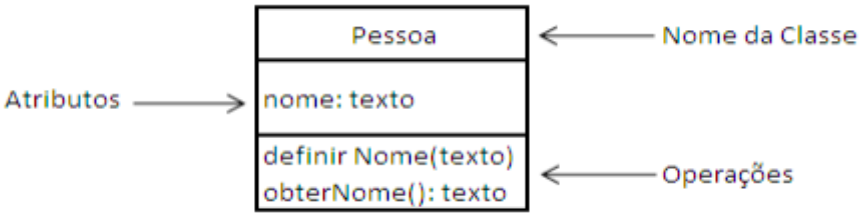


Figura 3 – Ilustração da declaração de Classes

Herança

Capacidade de uma classe herdar propriedades e código de uma outra classe da qual é descendente. Portanto, trata-se de um mecanismo que permite a reutilização de difinições e comportamentos que são adicionados, automaticamente, a novos objectos.

Neste caso, a classe herdada é denominada de superclasse e a classe herdeira denominada de subclasse .

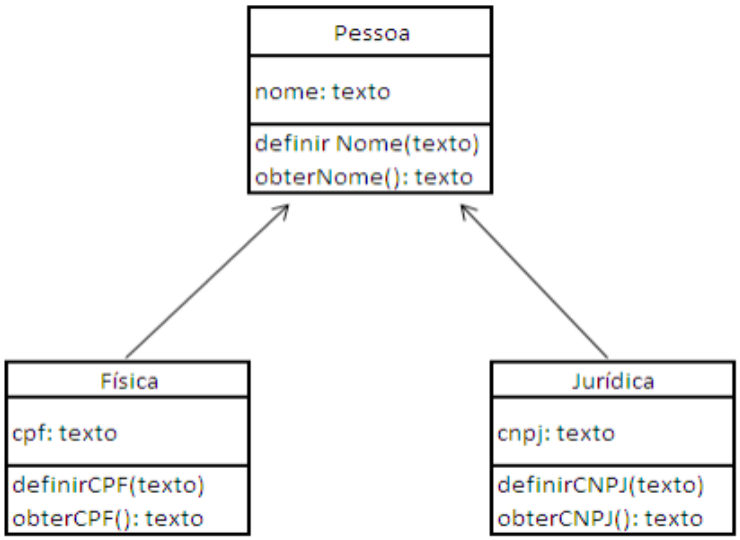


Figura 4 – Ilustração entre Herança e Classe.

Conclusão

Os Sistemas de Gestão de Banco de Dados Orientados a Objetos foram uma das grandes idéias do início dos anos 80. Contudo, até hoje, está em desenvolvimento e ainda não é um modelo definido, por isso pouco usado pelas empresas. No entanto, o surgimento cada vez maior de bancos de dados não convencionais para aplicações específicas aumenta o valor e o interesse para a tecnologia orientada a objeto. Nesta secção, procuramos abordamos conceitos fundamentais da BDOO, tais como: Identidade de Objecto; Estrutura do Objecto; Objecto; Objectos Complexos; Polimorfismo; Encapsulamento; Modelos de classes e Herança.

Avaliação da actividade

Verifique a sua compreensão!

1. Faça um resumo sobre as carateristicas da BDOO.
2. Dê um exemplo de um caso usando o conceito de herança.
3. Em que consiste o poliformismo numa BDOO.
4. Em que consiste o termo Poliformismo?
5. Dê um exemplo de poliformismo relacionado com o sistema educacional.
6. Qual o motivo da criação do modelo orientado a objectos?

Actividades de Aprendizagem

Actividade 1.2. Base de Dados Objeto-Relacional e Relacional-Estendido

Contextualização

A área de actuação dos SGBDs Objecto-Relacional tenta suprir a dificuldade dos sistemas relacionais convencionais, que é o de representar e manipular dados complexos. A solução proposta é a adição de facilidades para manusear tais dados utilizando-se das facilidades SQL existentes. Para isso foi necessário adicionar: extensões dos tipos básicos no contexto SQL; representações para objectos complexos no contexto SQL; herança no contexto SQL; sistema para produção de regras.

Base de dados Objecto-Relacional

Nesta secção falaremos do modelo de dados objecto-relacional (BDOR) ou (SGBDRO). Trata-se de um sistema de gestão de base de dados (SGBD) semelhante a uma base de dados relacional, porém com um modelo de base de dados orientado a objectos: objectos, classes e herança são suportados diretamente nos esquemas do banco de dados e na linguagem de consulta. Além disso, ele suporta extensão do modelo de dados com a personalização de tipos de dados e métodos.

Este modelo de base de dados surgiu

numa tentativa de unir as vantagens de ambas as tecnologias anteriores. As BDOR modelam objectos armazenados em tabelas, ou seja, utilizam as tabelas do modelo relacional, mas nelas são armazenados objectos, com seus atributos e comportamentos, unindo assim os paradigmas.

Ou seja, os dados são armazenados em relações, mas pode-se armazenar dados complexos abstraindo seu comportamento da mesma forma como é feito na orientação a objectos. Na prática, cria-se um tipo X (que pode ser visto como objecto nesse contexto) que tem n atributos (ou colunas); e depois ao se criar uma tabela A, uma das suas colunas será do tipo X, e nessa coluna, ao invés de se armazenar um tipo comum de dados (int, varchar, decimal, etc.) se armazena o tipo X que por sua vez poderá ter várias colunas de tipos comuns ou mesmo outros tipos.

Esse quadro geral adquire assim uma aparência de objectos e suas características, ou ao menos, de dados complexos, o que já é um avanço em relação ao tradicional modelo relacional. Como exemplo de aplicações que se utilizam dessa tecnologia destacam-se: softwares de armazenamento de imagens obtidas por satélite, projetos de arquitetura, mapas geoespaciais e de relevo, dentre outros.

Porque utilizar a BDOR?

Alguns motivos que levam à utilização desse tipo de modelo são:

- A incapacidade do modelo relacional básico de resolver os desafios e atender as necessidades das novas aplicações;
- A capacidade de armazenar novos tipos de dados;
- Fornecem suporte para consultas complexas sobre dados complexos;
- Atendem aos requisitos das novas aplicações e da nova geração de aplicações de negócios;
- Algumas aplicações para as quais é necessário o uso desse tipo de modelo são:
- Armazenamento de imagens (obtidas por satélite ou de alguma outra forma digital);
- Dados complexos não-convencionais em projeto de engenharia;
- Grandes informações sobre o genoma biológico;
- Projetos de arquitetura;
- Dados de séries temporais em transações;
- Dados sobre o espaço (regiões geográficas, criação de mapas);
- Sistemas móveis e distribuídos, dentre outros;
- Algumas base de dados que fazem uso do modelo objeto-relacional são Oracle 9i, DB2/6000, ILUSTRA, UniSQL, PostegresSQL/Postegres e CA-Ingres.

Evolução e tendências actuais da tecnologia de base de dados

No mundo comercial actual, há várias famílias de produtos de SGBDs disponíveis. As duas mais importantes são as de SGBDRs e SGBDOs, que se baseiam nos modelos de dados relacional e de objetos, respectivamente. Os dois outros tipos principais de produtos de SGBDs — hierárquicos e de rede — são actualmente referenciados como SGBDs legados; eles são baseados nos modelos de dados hierárquico e de rede, que foram introduzidos na metade dos anos 60.

A família dos hierárquicos tem um produto predominante — o IMS da IBM, enquanto a família dos de rede abrange um extenso número de SGBDs, como o IDS II (Honeywell), IDMS (Computer Associates), IMAGE (Hewlett Packard), VAX-DBMS (Digital) e TOTAL/SUPRA (Cincom). À medida que a tecnologia evolui, os SGBDs legados são gradualmente substituídos pelos novos que surgem. Nesse contexto, devemos focar o grave problema da interoperabilidade — a interconexão entre várias base de dados pertencentes às mais diversas famílias de SGBDs — como também com sistemas (legados) de gerenciamento de arquivos.

Toda uma série de novos sistemas e ferramentas surgiu para lidar com esse problema. Mais recentemente, o XML despontou como um novo padrão para troca de dados na WEB. O principal impulso para o desenvolvimento de SGBDORs estendidos surgiu da não-aderência dos SGBDs legados e do modelo relacional básico — assim como dos primeiros SGBDRs — para atender às demandas das novas aplicações, que estão basicamente em áreas que

envolvem uma variedade de tipos de dados — por exemplo, textos em publicação eletrônica, imagens em fotos de satélite ou em previsão do tempo, dados complexos não convencionais em projetos de engenharia, em informações do genoma biológico ou em projetos arquiteturais; em séries de dados históricos de transações de mercado de ações ou histórico de vendas e dados espaciais e geográficos em mapas, dados sobre poluição da água ou do ar, ou informações sobre tráfego.

Assim, há uma clara necessidade de se projetar base de dados que possam desenvolver, manipular e manter objectos complexos que surgiram com as novas aplicações. Além disso, tornou-se necessário manipular informações digitalizadas que armazenem sequências de dados de áudio e vídeo (particionadas em quadros individuais) exigindo dos SGBDs o armazenamento de BLOBs (Binary Long Objects — Objetos Binários Longos).

A popularidade do modelo relacional foi favorecida por uma infra-estrutura muito robusta em termos de SGBDs comerciais, que foram projetados para suportá-lo. Porém, o modelo relacional básico e as primeiras versões da linguagem SQL mostraram-se inadequados para atender aos novos desafios.

Os modelos de dados legados, como o modelo de rede, possuem a facilidade de modelar relacionamentos explicitamente, mas falham pelo uso intenso de ponteiros na implementação e não possuem conceitos como identidade de objetos, herança, encapsulamento e suporte a diversos tipos de dados e objetos complexos.

O modelo hierárquico é bem adequado a algumas ocorrências típicas de hierarquias no mundo real e nas organizações, mas é muito limitado e rígido em termos de caminhos hierárquicos internos aos dados. Assim, iniciou-se uma tendência de se combinar as melhores características do modelo de dados e das linguagens de objetos com o modelo de dados relacional, de modo que ele possa ser estendido para lidar com os desafios das aplicações de hoje.

Implementação e aspectos relacionados a sistemas de tipos estendidos

Existem várias questões de implementação associadas ao suporte de um sistema de tipos estendidos com funções associadas (operações). Nesse momento, resumiremos brevemente essas questões.

- Os SGBDORs devem associar dinamicamente uma função definida pelo utilizador em seu espaço de endereçamento apenas quando este é chamado. Nos SGBDORs, várias funções são requeridas para operar sobre dados espaciais em duas ou três dimensões, imagens, textos, e assim por diante. Com ligação estática para toda as bibliotecas de funções, o espaço de endereçamento do SGBD pode aumentar em ordem de magnitude.

- Os aspectos cliente-servidor lidam com a substituição e a activação de funções. Se o servidor precisar executar um função, é preferível fazê-la no espaço de endereçamento do SGBD em vez de remotamente, em razão do alto volume de sobrecarga. Caso a função demande uma computação muito intensiva ou se o servidor estiver atendendo um grande número de clientes, o servidor pode transferir a função para uma máquina cliente isolada. Por questões de segurança, é melhor executar as funções no cliente utilizando o ID de utilizador do cliente. Para futuras funções, é desejável que sejam escritas em linguagens interpretadas como JAVA.
- Deve ser possível efectuar consultas dentro de funções. Uma função deve operar do mesmo modo quando é utilizada; a partir de uma aplicação por meio de uma API (Aplication Program Interface) ou quando é chamada pelo SGBD como parte de uma execução SQL com a função embutida em um comando SQL. Os sistemas devem suportar um aninhamento desses callbacks.
- Em razão da variedade de tipos de dados em um SGBDOR e das operações associadas, a eficiência no armazenamento e no acesso aos dados é importante. Para dados espaciais ou multidimensionais, podem ser utilizadas novas estrutura de armazenamento, como árvores-R (R_trees) e árvores quad (quad trees) ou arquivos Grid. Os SGBDORs devem permitir a criação de novos tipos com novas estruturas de acesso. Ao lidar com cadeias de texto longas ou arquivos binários também se apresentam várias opções de busca e armazenamento. Deve ser possível explorar essas novas opçõe pela definição de novos tipos de dados no SGBDOR.

Conclusão

Nesta actividade abordamos o modelo objecto-relacional e como o seu nome diz, estamos falando de um modelo relacional e um modelo orientadoa objectos. Em seguida mostramos algumas dos motivos da sua utilização tendo como destaque a incapacidade do modelo relacional. também descutimos a história e as tendências em sistemas de gestão de base de dados que se desenvolveram em direção a SGBDs objeto-relacionais (SGBDORs), assim como apresentamos as questões implementação e aspectos relacionados a sistemas de tipos estendidos.

Avaliação da actividade

Verifique a sua compreensão!

1. Qual é a necessidade de utilizar o modelo O-R?
2. Comente sobre uma das implementação associadas ao suporte de um sistema de tipo estendidos.
3. Dê exemplo de três base de dados O-R.

Actividades de Aprendizagem

Actividade 1.3. Linguagem e desenho de Base de Dados

Contextualização

Existem pelo menos dois factores que levam a adoção desse modelo:

- O primeiro é que na BD relacional se torna difícil trabalhar com dados complexos.
- A segunda é que aplicações são construídas em linguagens orientadas a objectos (java, C++, C#) e o código precisa ser traduzido para uma linguagem que o modelo de BD relacional entenda, o que torna essa tarefa muito difícil. Essa tarefa também é conhecida como "perda por resistência". (ELMASRI, 2005)

O modelo OO ganhou espaço nas áreas como base de dados espaciais, telecomunicações, e nas áreas científicas como física de alta energia e biologia molecular. Isso porque essa tecnologia oferece aumento de produtividade, segurança e facilidade de manutenção.

Como objectos são modulares, mudanças podem ser feitas internamente, sem afectar outras partes do programa. O modelo OO não teve grandes impactos nas áreas comerciais embora tenha sido aplicado em algumas. Em 2004 as BDOO tiveram um crescimento devido ao surgimento de BDOO livres.

ODMG

A Object Data Management Group (ODMG) e a Object Query Language (OQL) padronizaram uma linguagem de consulta para objectos. Uma característica que vale a pena ser ressaltada, é que o acesso a dados pode ser bem mais rápido, porque não são necessárias junções. Já que o acesso é feito diretamente ao objecto seguindo os ponteiros.

Outra característica importante é que o BDOO oferece suporte a versões, isto é, um objecto pode ser visto de todas e várias versões. As BDOO e relacionais apresentam uma série de características, e cada uma tem a sua vantagem e desvantagem.

- Como por exemplo, os modelos OO utilizam interfaces navegacionais ao invés das relacionais, e o acesso navegacional é bem eficiente implementada por ponteiros.
- Um problema seria a inconsistência desse modelo em trabalhar com outras ferramentas como OLAP, backup e padrões de recuperação. E os críticos afirmam que o modelo relacional é fortemente baseado em fundamentos matemáticos o que facilita a consulta, já os modelos OO não, o que prejudicaria e muito as consultas.
- A dificuldade de implementar encapsulamento seria um outro problema, porque como seriam feitas as consultas se não é possível ver os atributos?

Object Definition Language (ODL) – Linguagem de Definição de Dados

A ODL foi feita para dar suporte aos construtores semânticos do modelo de objectos ODMG e é independente de qualquer linguagem de programação em particular. O objetivo da ODL é criar especificações de objectos, isto é, classes e interfaces.

A ODL não é considerada uma linguagem de programação completa. Ela permite que o utilizador especifique um banco de dados independente da linguagem de programação, e utilizando o binding específico com a linguagem para especificar como os componentes ODL podem ser mapeados para componentes em linguagens de programação específica, como C++, SmallTalk e Java.

Language (OQL) – Linguagem de Consulta de Objectos

É a linguagem de consulta declarativa definida pela ODMG (1995). Prevê suporte ao tratamento de objectos complexos, invocação de métodos, herança e polimorfismo. É projetada para trabalhar acoplada com as linguagens de programação com as quais o modelo ODMG define um biding, como C++, SMALLTALK e JAVA. Isso faz com que qualquer consulta OQL embutida em uma dessas linguagens de programação pode retornar objectos compatíveis com os sistemas de tipos dessa linguagem.

Fornece suporte para lidar com set, structure, list e array, tratando estas construções com a mesma eficiência. Permite expressões aninhadas. Pode chamar métodos dos tipos envolvidos na consulta. Não fornece operadores para atualização, mas pode chamar operações definidas nos objectos para realizar esta tarefa, não violando assim a semântica do modelo de objectos, o qual, por definição, é gerenciado pelos métodos especificados no objecto. É possível definir um nome para uma determinada consulta, que é armazenada no BD.

Para uso posterior, a consulta é referenciada através do nome definido. Apresenta construtores de objectos, structure, set, list, bag e array. Uma consulta em OQL parte dos pontos de entrada do banco de dados e constrói como resposta, um objecto que é tipicamente uma coleção. Suporta as cláusulas SELECT, FROM, WHERE, GROUP BY, HAVING e ORDER BY.

Conclusão

É possível implementar um sistema de complexo usando em um SGBD Orientado a Objectos. A manipulação de objectos nativamente, aumenta a performance e os ganhos de desempenho de linguagens orientadas a objecto como o Java e as linguagens da plataforma NET. Permite maior desempenho e redução no tempo de desenvolvimento do software, já que não é necessário traduzir o modelo orientado a objeto para um modelo relacional, eliminando a complexidade extra e a perda de performance com a conversão para outros formatos como SQL.

Avaliação da actividade

Verifique a sua compreensão!

1. Por que razão o modelo OO deve estar associado as linguagens de programação OO?
2. Porque que muitas empresas não optam por este modelo OO?
3. Qual foi a necessidade de se padronizar uma linguagem de consulta para objectos?

Resumo da Unidade 1

O paradigma da orientação a objectos possui uma maneira própria de representar um problema. Essa maneira difere muito da forma tradicional de modelagem de dados utilizada pelas BD relacionais, apesar de apresentar algumas semelhanças, sobretudo, relativas à cardinalidade das relações entre as entidades.

Superficialmente, pode-se dizer que orientação a objectos corresponde à organização de sistemas como uma coleção de objectos que integram estruturas de dados e comportamento. Além desta noção básica, a abordagem inclui um certo número de conceitos, princípios e mecanismos que a diferenciam das demais.

Uma vantagem desse modelo é de não precisar de um DBA, pois sua administração é da responsabilidade do próprio analista de sistemas. Por ser uma tecnologia nova, muitas empresas preferem não arriscar, pois o modelo relacional ainda é muito empregue nos dias actuais. Migrar de uma tecnologia bem difundida no mercado para uma que está apenas começando seria muito arriscado. Essa é uma das desvantagens do BDOO que ocasiona poucas aplicações nessa nova tecnologia. Muitos ainda não confiam na sua integridade. À medida que a complexidade for aumentando, as empresas vão cada vez mais buscar por alternativas que consigam adequar às suas necessidades.

Avaliação da Unidade 1

Verifique a sua compreensão!

1. Qual é a filosofia do surgimento do modelo OR?
2. Porque utilizar a BDOR?
3. Defina Objecto complexo não-estruturado?

Soluções

1. O modelo de base de dados surgiu numa tentativa de unir as vantagens de ambas as tecnologias anteriores, as BDOR modelam objectos armazenados em tabelas, ou seja, utilizam as tabelas do modelo relacional, mas nelas são armazenados objectos, com seus atributos e comportamentos, unindo assim os paradigmas.
2. Alguns motivos que levam à utilização da BDOR são: A incapacidade do modelo relacional básico de resolver os desafios e atender as necessidades das novas aplicações; A capacidade de armazenar novos tipos de dados; Fornecem suporte para consultas complexas sobre dados complexos e atendem aos requisitos das novas aplicações e da nova geração de aplicações de negócios.
3. É um objecto que utiliza um tipo que não é padrão na BD e pode ocupar uma grande área de armazenamento. Estes objectos são do tipo BLOB, que significa Binary Large Object, e podem armazenar, por exemplo, os dados referentes a uma imagem bitmap, um texto formatado, uma música entre outros.

Leituras e outros Recursos

Documentos acedidos via Internet no dia 27.02.2016

https://pt.wikipedia.org/wiki/Banco_de_dados_objeto-relacional

http://repositorio.ufla.br/bitstream/1/8354/1/MONOGRAFIA_Banco_de_dados_relacional_e_objeto_relacional_uma_compara%C3%A7%C3%A3o_usando_postgresql.pdf

https://pt.wikipedia.org/wiki/Banco_de_dados_orientado_a_objectos

http://www.fsma.edu.br/si/edicao3/banco_de_dados_orientado_a_objectos.pdf

http://www.fatecsp.br/dti/tcc/tcc0002.pdf

Unidade 2. Processamento de consultas e Transações em Base de Dados

Introdução à Unidade

O processamento de consultas é a actividade responsável em extrair informações de uma base de dados. Quando uma consulta é submetida ao sistema, ela deve ser traduzida e transformada em uma sequência de operações que possa ser efectivamente executada pelo sistema computacional. No entanto, essa tradução permite que o sistema obtenha uma sequência de comandos optimizados que possibilitam o processamento da consulta de uma forma eficiente. Ainda nesta unidade, abordaremos o conceito de transações pois trata-se de um assunto indispensável em base de dados, pois este termo refere-se a um conjunto de operações que formam uma única unidade de trabalho lógico. Por exemplo a transferência de dinheiro de uma conta para a outra é uma transacção consistindo em duas actualizações, uma para cada conta.

Objetivos da Unidade

Após a conclusão desta unidade, deverá ser capaz de:

1. Descrever uma consulta e processamento de consulta;
2. Descrever o processo de consulta e optimização de consulta;
3. Descrever uma transação;
4. Exemplificar casos de transações;
5. Distinguir os conceitos commit, rollback e checkpoint.

Termos-chave

Consulta, Processamento de consulta, Optimização de consulta, Transação, Commit, Rollback e Checkpoint.

Actividades de Aprendizagem

Actividade 2.1. Processamento de Consultas

Contextualização

Uma consulta (query) é uma especificação de alto nível de um conjunto de objectos de interesse da base de dados . Geralmente é especificada em uma linguagem especial (Data Manipulation Language - DML) que descreve o que o resultado deve conter .

Optimização de consulta (Query optimization) tem o objetivo de escolher uma estratégia de execução eficiente para a execução da consulta.

Processamento de consulta (Query processing) em base de dados orientados a objecto é quase o mesmo que no banco de dados de relação com apenas algumas mudanças por causa das diferenças semânticas entre consultas relacionais e orientados a objecto. Além disso, todas as técnicas aplicáveis a um são também para outro. De facto, sem a hierarquia de classes, as consultas têm estruturas semelhantes em ambas as bases de dados. Portanto, a fim de simplificar, vamos ver processamento de consultas, sem distinção entre base de dados de objetos e base de dados relacionais.

A Consulta é processada em duas fases: a fase de optimização de consulta e a fase de processamento da consulta. Para facilitar a compreensão, vamos adicionar a fase de consulta de compilação antes de as duas fases anteriores porque as consultas são vistas pelo utilizador como Data Manipulation Language scripts (DML). Assim, a figura abaixo apresenta todo o processo .

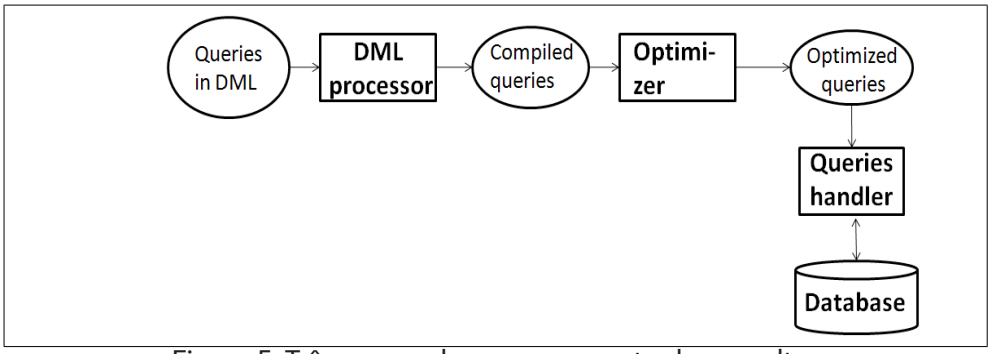


Figure 5. Três passos do processamento de consultas

a. Compilação de consulta (Query compilation): processador DML traduz instruções DML em instruções de baixo nível (de consulta compilado) que o optimizador de consulta pode entender.

b. Optimização de consulta (Query optimizer): o optimizador gera automaticamente um conjunto de estratégias razoáveis para processar uma determinada consulta , e selecciona um ideal em função do custo esperado de cada uma das estratégias geradas.

c. Processamento de consulta (Query Processing): o sistema executa a consulta utilizando a estratégia ideal gerada.

Na optimização de consulta, uma consulta SQL é primeiro traduzida em uma expressão de álgebra relacional equivalente, usando uma estrutura de dados árvore de consulta, antes de ser optimizado. Vamos, portanto, definir, a seguir, como passar de uma consulta SQL para uma expressão em álgebra relacional .

Traduzindo consultas SQL em Álgebra Relacional

Para traduzir consulta SQL em álgebra relacional, precisamos conhecer os diferentes operadores em ambos e a correspondência entre eles. No módulo intitulado " Princípio de Sistemas de Base de Dados", a álgebra relacional tem sido apresentada em detalhe. Então , não vamos fazê-lo novamente .

Como na optimização de consulta, uma consulta SQL é decomposto em partes da consulta também chamados de unidades elementares, cada parte tem que ser expressa por operadores algébricos e optimizado. Uma unidade de consulta é uma única expressão SELECT- FROM-WHERE -GROUP BY- HAVING. Desde as duas últimas cláusulas (GROUP BY e HAVING) não são necessariamente usados em uma consulta, estas cláusulas não costumam aparecer na unidade de consulta. Consulta tendo as consultas aninhadas é decomposto em unidades de consulta separados .

Exemplo:

Consulta inicial	Decomposição
SELECT name FROM clients WHERE address = (SELECT address FROM clients WHERE numClient=76);	subquery = SELECT address FROM clients WHERE numClient=76; SELECT name FROM clients WHERE address = subquery;

Tabela 1: Decomposição de consulta SQL.

Entre as estratégias geradas pelo optimizador de consulta para cada unidade, uma delas é escolhida.

Existem dois tipos de consultas aninhadas : não correlacionadas e correlacionadas.

Consultas aninhadas não correlacionadas poderiam ser realizadas separadamente e seus resultados seriam utilizados na consulta externa.

Consultas aninhadas correlacionados precisam de informação (variável tupla) a partir de consulta externa na sua execução.

As primeiras são mais fáceis de optimizar em comparação com as últimas. A consulta externa de exemplo acima é não correlacionadas, pois ela pode ser realizada independentemente da outra.

A tabela a seguir descreve o mapeamento dos operadores SQL e operadores de álgebra relacional .

SQL	Álgebra Relacional
Select * from table ;	table(x1, x2, …, xn)
Select * from table1, table2 ;	table1(x1, x2, …, xn) ´ table2(y1, y2, …, ym)
Select Distinct x1, x2 from table;	Px1,x2table(x1, x2, …, xn)
Select * from table where x1 > x2;	sx1,x2table(x1, x2, …, xn)
Select * from table1 UNION Select * from table2;	table1(x1, x2, …, xn) È table2(y1, y2, …, ym)
Select x1, x2, x3 from table1 EXCEPT Select y1, y2, y3 from table2;	table1(x1, x2, x3) - table2(y1, y2, y3)
Select x1, x2, x3 from table1 INTERSECT Select y1, y2, y3 from table2;	table1(x1, x2, x3) Ç table2(y1, y2, y3)
Select * from table1, table2 where x1>y1;	qx1>y1(table1(x1, x2, x3) - table2(y1, y2, y3))
Select * from table1, table2 where x1=y1;	qx1=y1(table1(x1, x2, x3) - table2(y1, y2, y3))
Select count(*) from table group by x1;	Countx1(table(x1, x2, x3))
Select SUM(x2) from table group by x1;	Sumx1(table(x1, x2, x3), x2)

Tabela 2: Consultas SQL para Álgebra Relacional

Heuristica

A heurística é uma das opções que possibilita a optimização de uma consulta, ela utiliza regras que modificam a representação interna da consulta e melhoram o desempenho esperado para a execução dessa consulta.

Quando ocorre o uso da heurística, primeiramente, o parser de uma consulta de alto-nível gera a representação interna inicial seguindo de acordo com as regras pré- estabelecidas de heurísticas. Resumidamente, as regras de heurística são:

1. executar operações de seleção o mais cedo possível, para eliminar tuplas que não preenchem a condição de seleção;
2. combinar seleções com um produto cartesiano anterior, de forma a produzir uma junção natural, que é bem mais econômica do que um produto cartesiano;
3. aplicar operações de projeção de forma antecipada, isto é, transferir ou inserir projeções sempre que necessário;
4. aplicar operações em grupo à medida que se percorre as tuplas, pois evita a geração de muitas tabelas temporárias;
5. procurar expressões comuns em uma árvore, devendo computá-las uma única vez e guardá-las em uma tabela temporária [Elmasri e Navathe, 2000].

Depois que o parser gerou a representação interna inicial, um plano de execução de consulta é criado para executar grupos de operações baseados em caminhos de acesso disponíveis nos arquivos envolvidos na consulta. A árvore de consulta representa a entrada de relações da consulta através de nós e também representa as operações da álgebra relacional com nós internos. A execução da árvore de consulta consiste em executar operações de nós internos sempre que seus operandos estão disponíveis e substitui aquele nó interno pela relação resultante da execução da operação. A execução termina quando os nós da raiz são executados e produzem uma relação resultante para a consulta.

Por exemplo: para todo projecto localizado em 'Stafford', deverá ser recuperado o número do projecto, o número de controle do departamento, o último nome do gerente, o endereço e a data de aniversário. Esta consulta corresponde à seguinte expressão da álgebra relacional:

$$\pi_{\text{PNUMBER, DNUM, LNAME, ADDRESS, BDATE}}(((\sigma_{\text{PLOCATION = 'Stanfford'}}(^{\text{PROJECT}})) \; |\times|_{\text{D.NUM = DNUMBER}} (^{\text{DEPARTMENT}})) \; |\times|_{\text{MGRSSN = SSN}} (^{\text{EMPLOYEE}}))$$

que corresponde à seguinte consulta SQL:

```
Q2:  SELECT   P.PNUMBER, P.DNUM, E.LNAME, E. ADDRESS, E.BDATE
     FROM     PROJECT AS P, DEPARTMENT AS D, EMPLOYEE AS E
     WHERE    P.DNUM = D.DNUMBER AND D.MGRSSN = E.SSN AND
              P.PLOCATION = 'Stafford';
```

As figuras (6) e (7) ilustram duas árvores de consultas para uma consulta Q2. A figura (6) representa a árvore de consulta que corresponde à expressão da álgebra relacional de Q2 e a figura (6) ilustra a árvore de consulta para consultas SQL de Q2. A figura (8) mostra o grafo da consulta Q2.

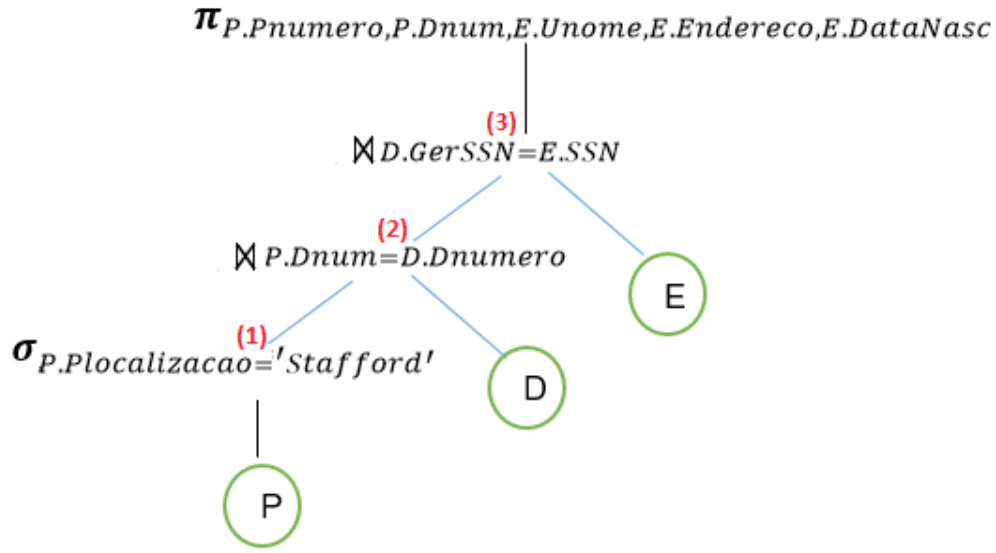


Figura 6 . Árvore de consulta em álgebra relacional da consulta Q2

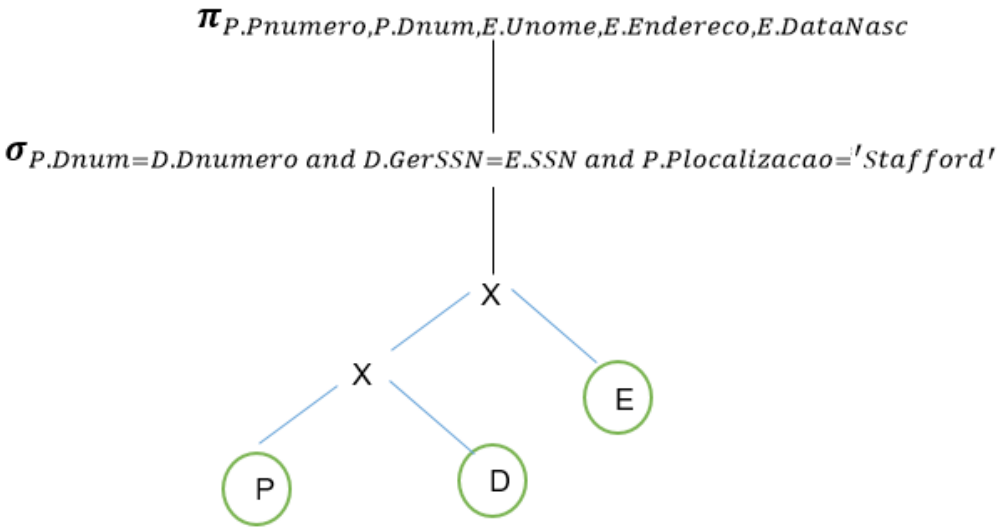


Figura 7. Àrvore de consulta em SQL da consulta Q2

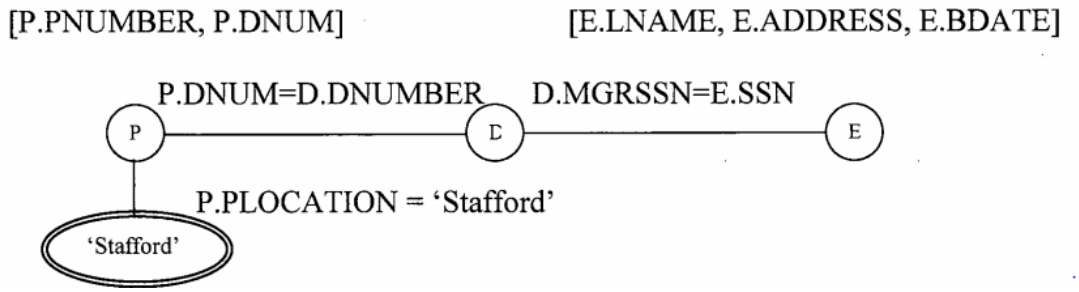


Figura 8. Grafo da consulta Q2.

Quando temos consulta de relações complexas, operações unárias (SELECT e PROJECT) devem ser realizadas em primeiro lugar. Depois operações binárias (produto, JOIN, UNION, DIFERENÇA ...) pode ser realizadas. Com efeito, o tamanho do resultado de uma operação binária é o produto dos tamanhos de dois operandos. Por outro lado, as operações unárias tendem a reduzir o tamanho do resultado . Por conseguinte , as operações unárias deverá ser aplicadas antes de qualquer operação binária.

Example: Considere as três relações:

Affectation (pCode, empNum, hour),

Employee (empNum, empName),

Job (position, rate).

A consulta a seguir pode ser efectuada em quatro maneiras descritas na tabela abaixo: Q = O nome e o número de funcionários que trabalham em projecto PR40 e com mais de 10000 como a taxa.

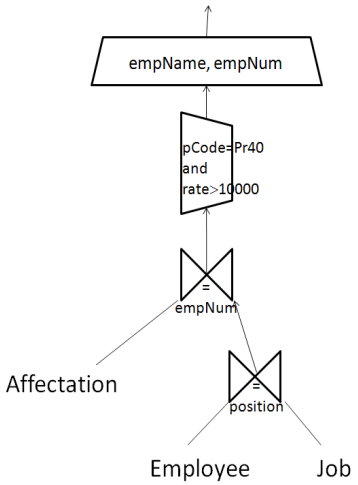
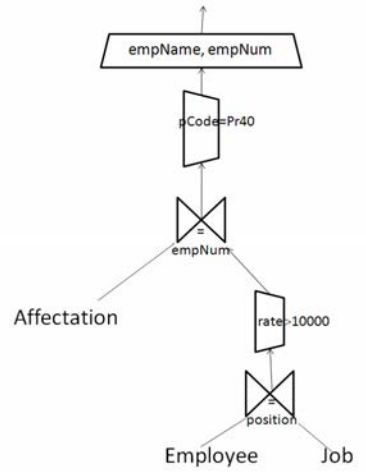
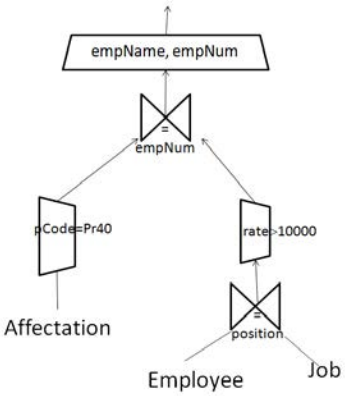
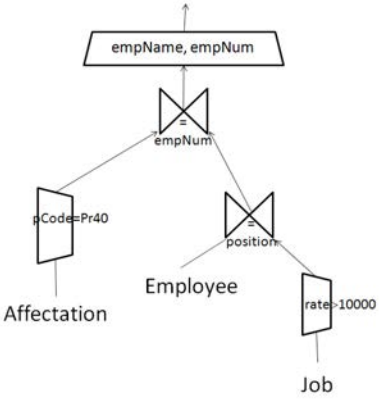
<table>
<tr><td>Consulta 1</td><td>Consulta 2</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Consulta 3</td><td>Consulta 4</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

Tabela 3: árvores de execução de uma consulta

Conclusão

Esta secção abordou o processamento de consulta (query processing). Apresentamos as diferentes fases do processo, os principais são a optimização de consultas e processamento de consultas. Além disso, traduzimos as consultas SQL em expressões de álgebra relacional. Por fim tratamos do conceito euristica usada optimização da consulta.

Avaliação da actividade

Verifique a sua compreensão!

1. O que entende por consulta?
2. Defina optimização de consulta.
3. Diferencie optimização de consulta de processamento de consulta.

Actividades de Aprendizagem

Actividade 2.2. Transacção em base de dados

Contextualização

A transacção consiste numa unidade lógica de trabalho. Se for composta por um conjunto de pessoas, então, ou se realizam todos ou não se realiza nenhum.

Do ponto de vista popular pode se dizer que transacção é um caso de tudo ou nada.

Os SGBD contém um mecanismo denominado "Gestão Transaccional" ou "controlo Transaccional", que permite que os dados sejam sempre processados dentro do contexto de transacções.

Mesmo que o objecto seja executado em uma única operação, esta terá de ser executada dentro de uma transaccão, formada por um único comando.

ACID

(Acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade - do inglês: Atomicity, Consistency, Isolation, Durability), é um conceito utilizado em ciência da computação para caracterizar, entre outras coisas, uma transação em uma Base de Dados.

Atomicidade: Trata o trabalho como parte indivisível (atômico). A transação deve ter todas as suas operações executadas em caso de sucesso ou nenhum resultado de alguma operação refletido sobre a base de dados em caso de falha. Ou seja, após o término de uma transação (commit ou abort), a base de dados não deve refletir resultados parciais da transação. Por exemplo:

1. Ou todo o trabalho é feito, ou nada é feito.
2. Em uma transferência de valores entre contas bancárias, é necessário que, da conta origem seja retirado um valor X e na conta destino seja somado o mesmo valor X. As duas operações devem ser completadas sem que qualquer erro aconteça, caso contrário todas as alterações feitas nessa operação de transferência devem ser desfeitas.

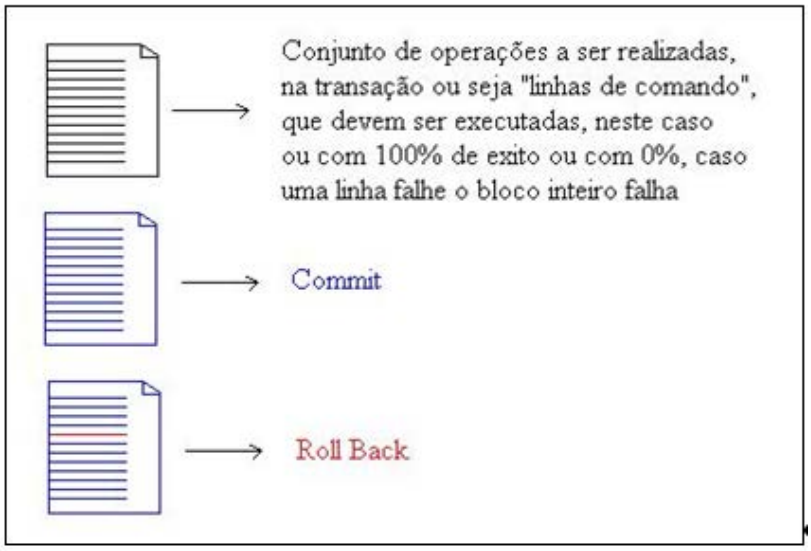


Figura 9. Transações de processos (Atomicidade)

Consistência: A execução de uma transação deve levar BD de um estado consistente a um outro estado consistente, ou seja, uma transação deve respeitar as regras de integridade dos dados (como unicidade de chaves, restrições de integridade lógica etc...). Por exemplo: considere uma BD que guarde informações de clientes e que use o CPF como chave primária. Então, qualquer inserção ou alteração no banco não pode duplicar um CPF (unicidade de chaves) ou colocar um valor de CPF inválido, como o valor 000.000.000-00 (restrição de integridade lógica).

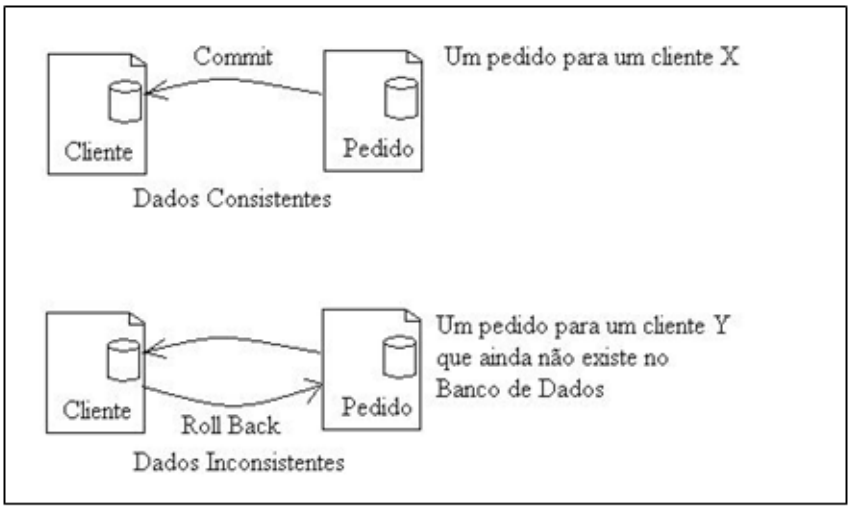


Figura 10. Transações de processos (Consistência)

Isolamento: Em sistemas multi utilizadores, várias transações podem estar acessando o mesmo registro (ou parte do registro) na BD. Por exemplo, se um utilizador tentasse alterar um registro e um outro estivesse tentando ler este mesmo registro.

O isolamento é um conjunto de técnicas que tentam evitar que transações paralelas interfiram umas nas outras, fazendo com que o resultado de várias transações em paralelo seja o mesmo resultado destas mesmas transações sendo executadas sequencialmente (uma após a outra). Operações exteriores a uma dada transação jamais verão esta transação em estados intermediários.

Durabilidade: Os efeitos de uma transação em caso de sucesso (commit) devem persistir na Base de dados mesmo em casos de quedas de energia, travamentos ou erros. Garante que os dados estejam disponíveis em definitivo.

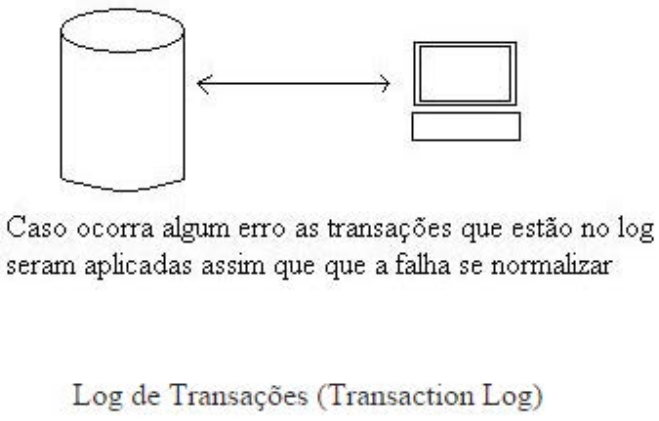


Figura 11. Transações de logs (Consistência)

Conclusão

Esta secção abordou o conceito de transações, portanto faça um resumo dos conceitos aqui abordados.

Avaliação da actividade

Verifique a sua compreensão!

1. O que são transações?
2. Diferencie commit de rollback.
3. Em que consiste a atomicidade?

Actividades de Aprendizagem

Actividade 2.3. Operações em Transacções de Base de Dados

Contextualização

Um conjunto de várias operações em um BD pode ser vista pelo utilizador como uma única unidade. Por exemplo: A transferência de fundos de uma conta corrente pra uma conta poupança é uma operação única do ponto de vista do cliente, porém, dentro do sistemas de base de dados, ela envolve várias operações. É essencial que todo o conjunto de operações seja concluído, ou que, no caso de uma falha, nenhuma delas ocorra. Essas operações, que formam uma única unidade lógica de trabalho são chamadas de transações.

Operações de Leitura/Escrita

As operações de acesso a BD que uma transação pode fazer são:

- Read(X) que transfere o item de dados (X) da BD para um buffer local alocado à transação que executou a operação de read. O valor de X é colocado dentro de uma variável de programa.
- Write(X): que transfere o item de dados (X) do buffer local da transação que executou o write de volta para a BD. O valor da variável de programa é passado para o item de dado na BD. OBS.: para fins de simplificação, o item de dados e a variável de programa correspondente serão representados da mesma forma, (X).

Execução das Operações

Executar read(x):

- Encontrar o endereço do bloco de disco que contém (X);
- Copiar este bloco de disco para dentro do buffer/memória principal, se ele já não estiver lá;
- Copiar o item (X) do buffer para a variável de programa.

Executar write(x):

- Encontrar o endereço do bloco de disco que contém (X);
- Copiar o bloco de disco para a memória principal se ele já não estiver lá;
- Copiar o item (X) da variável de programa (X) para a localização correta no buffer; Copiar o bloco alterado do buffer de volta para o disco (imediatamente ou mais tarde).

Exemplo de Transação

read(x);	read(x);
x := x – N;	x:= x + M;
write(x);	write(x);
read(y);	T 2
y := y + N;	
write(y);	
T1	

Acesso concorrente ao dado (X).

Outro Exemplo

T1:

read(A);

A := A - 50; write(A);

read(B);

B := B + 50; write(B);

Seja T1 uma transação que transfere 50.00 da conta A para a conta B. Essa transação pode ser definida como:

Operações Adicionais

Uma transação é uma unidade de trabalho atômica que é completada em sua totalidade ou nada dela deve ter efeito. Para tornar isso possível, as seguintes operações são necessárias:

- Begin-transaction: Denota o início da execução da transação;
- End-transaction: Especifica que as operações da transação terminaram e marca o limite final da execução da transação. Neste ponto é necessário verificar se o COMMIT ou o ABORT são necessários, caso já não tenham sido explicitados; Commit-transaction: Sinal de término com sucesso e que as alterações podem ser "permanentemente" gravadas no BD;
- Rollback (abort-transaction): Assinala que a transação não terminou com sucesso e que seus efeitos devem ser desfeitos;
- Undo: Desfaz uma operação;
- Redo: Refaz uma operação.

Exemplo com Operadores (1)

```
Procedure Transfer;
begin
        start;  (begin transaction)
        input (FROMaccount, TOaccount, amount);
        temp := read(Accounts[FROMaccount]);
        if temp < amount then
                begin
                        output("insufficient funds");
                        Abort;
                end
                else
                begin
                        write(Accounts[FROMaccount],temp - amount);
                        temp = read(Accounts[TOaccount]);
                        write(Accounts[TOaccount], temp + amount);
                        commit;
                        output("transfer complete");
        end;
        return; (end transaction)
end.
```

Exemplo com Operadores (2)

```
begin transaction

update TABELA1 set CAMPO1 = 'NOVO VALOR'
where CAMPO2 = 35
if @@ERROR <> 0
rollback
else
commit

end transaction
```

Suponha uma tabela que possua 6000 tuplas que satisfaçam a condição e que na tupla 4666 houve uma falha no sistema. Neste caso, a variável chamada @@ERROR guardará o código do erro, o comando ROLLBACK será executado e as 4666 tuplas já actualizadas serão automaticamente voltadas a seu estado anterior. Ao contrário, se tudo correr bem, o comando COMMIT será executado e todas as 6000 tuplas serão actualizadas com êxito.

Estados de uma Transação

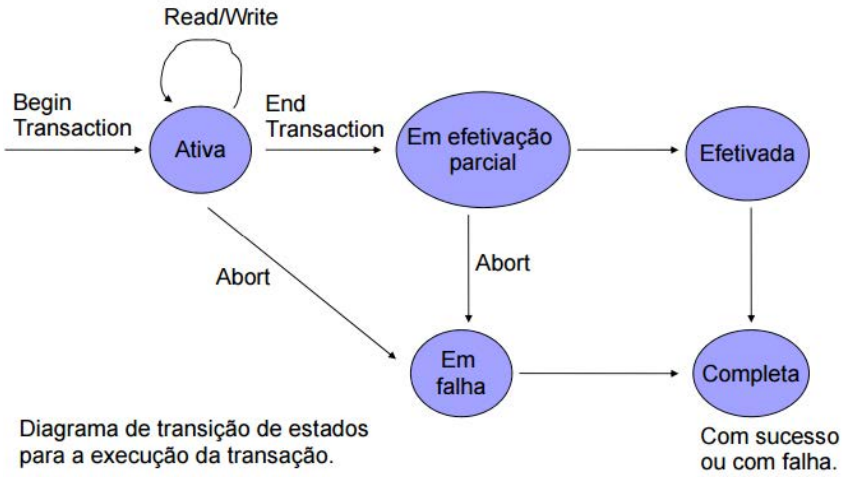


Figura 12. Estados de uma Transação

Conclusão

Uma transação entra no estado de falha quando o sistema determina que ela já não pode prosseguir sua execução normal. Essa transação deve ser desfeita e entra no estado de abortada. Nesse momento o sistema tem duas opções:

- Reiniciar a transação: somente se ela foi abortada como resultado de algum erro de hardware ou de software não criado pela lógica interna da transação. Uma transação reiniciada é considerada uma transação nova.
- Matar a transação: normalmente, isto é feito em decorrência de algum erro lógico interno e só pode ser corrigido refazendo o programa de aplicação. Esse erro dá-se ou porque a entrada de dados não era adequada ou porque os dados desejados não foram encontrados no banco de dados.

Avaliação da actividade

Verifique a sua compreensão!

1. Descreva a operação de leitura e escrita.
2. O que é feito quando é detectado um erro numa transacção?
3. Qual é a diferença entre commit e undo?

Conclusão da Unidade 2

Esta secção abordou o processamento de consulta (query processing) e o conceito de transações, sendo estudadas as diferentes fases do processo. As principais fases são a optimização de consultas e processamento de consultas. Além disso, traduzimos as consultas SQL em expressões de álgebra relacional. Por fim tratamos do conceito euristica usada optimização da consulta.

Avaliação da Unidade

Verifique a sua compreensão!

1. O que entendes por consulta?

   Sol. Uma consulta (query) é uma especificação de alto nível de um conjunto de objectos de interesse da base de dados. Geralmente é especificada em uma linguagem especial (Data Manipulation Language - DML) que descreve o que o resultado deve conter .

2. Defina optimização de consulta.

   Sol. Optimização de consulta (Query optimization) tem o objectivo de escolher uma estratégia de execução eficiente para a execução da consulta.

3. O que entendes por heurística?

   Sol. A heurística é uma das opções que possibilita a optimização de uma consulta, ela utiliza regras que modificam a representação interna da consulta e melhoram o desempenho esperado para a execução dessa consulta.

Unidade 3. Data Warehouse e Data Mining

Introdução à Unidade

Nos dias de hoje o sucesso das organizações está na sua capacidade de analisar, planear e reagir imediatamente, às mudanças nas condições dos seus negócios. Para que isso aconteça, é necessário que a organização disponha de mais e melhores informações, que constituem, reconhecidamente, a base destes processos. Os avanços da tecnologia de informação e comunicação vieram garantir a possibilidade das empresas manipularem grandes volumes de dados e atingirem um alto índice de troca de informações, com o uso das redes viabilizando operações em nível mundial.

Com a globalização, os negócios não têm mais fronteiras, ter a informação correcta no menor tempo possível e utilizar-se de sistemas de apoio à decisão tornou-se o grande diferencial para as empresas. Diariamente, dados sobre os mais variados aspectos dos negócios da organização são gerados e armazenados, e passam a fazer parte dos recursos de informação da mesma.

Por esse motivo, um novo conjunto de conceitos e ferramentas vem ganhando enorme destaque nos últimos anos como a tecnologia de Data Warehouse, que oferece às organizações uma maneira flexível e eficiente de obter as informações necessárias a seus processos decisórios, o conceito de Data Mining aos poucos ganha o seu espaço devido à necessidade de se filtrar as informações e, as base de dados distribuídas ganharam espaço no mercado com o surgimento das redes dos computadores e das aplicações Web capazes de gerir grande volume de informação.

Objetivos da Unidade

Após a conclusão desta unidade, deverá ser capaz de:

1. Descrever as características de um data werehouse e data mining;
2. Utilizar ferramentas para extrair informação em DW.
3. Descrever uma base de dados distribuída.
4. Listar as vantagens de uma base de dados distribuída.

Termos-chave

Base de Dados Distribuida, Redes de Computadores, Data mining, Data warehouse.

Actividades de Aprendizagem

Actividade 3.1: Base de Dados Distribuídas

Contextualização

Num sistema de base de dados centralizado, os dados, as aplicações que os manipulam e todo o processamento estão localizados no mesmo computador.

Todos sabemos que, apesar dos baixos custos de material informático, os recursos continuam a ser limitados e finitos. Não existem computadores com espaço infinito em disco ou memória. Por isso, por vezes, a solução passa por interligar diferentes computadores, partilhando alguns dos recursos, hardware e software. Devido à limitação local de cada computador, surgiu a necessidade de se partilhar recursos em rede de computadores e um desses recursos pode ser a base de dados, eis o surgimento das base de dados distribuídas.

Modelo Distribuído

O Sistema de Gestão de Base de Dados Distribuídos (SGBDD) controla o armazenamento e processamento de dados relacionados logicamente por meio de sistemas computacionais interconectados através de uma rede, em que tanto os dados como as funções de processamento são distribuídos entre os diversos locais.

Existem dois tipos de base de dados distribuídos, as homogéneas e as heterogéneas. As homogéneas são compostos pelas mesmas base de dados, já as heterogéneas são aquelas que são compostas por mais de um tipo de base de dados.

Numa base de dados distribuída os arquivos podem estar replicados ou fragmentados, esses dois tipos podem ser encontrados ao longo de seus nós.

Quando os dados se encontram replicados, existe uma cópia de cada um dos dados em cada nó, tornando as bases iguais (ex: tabela de produtos de uma grande loja).

Já na fragmentação, os dados se encontram divididos ao longo do sistema, ou seja a cada nó existe uma base de dados diferente, se olharmos de uma forma local, mas se analisarmos de uma forma global os dados são vistos de uma forma única, pois cada nó possui um catálogo que contém cada informação dos dados dos base de dados adjacentes.

A replicação dos dados pode se dar de maneira síncrona ou assíncrona.

- No caso de replicação síncrona, cada transacção é dada como concluída quando todos os nós confirmam que a transacção local foi bem sucedida.
- Na replicação assíncrona, o nó principal executa a transacção enviando confirmação ao solicitante e então encaminha a transacção aos demais nós.

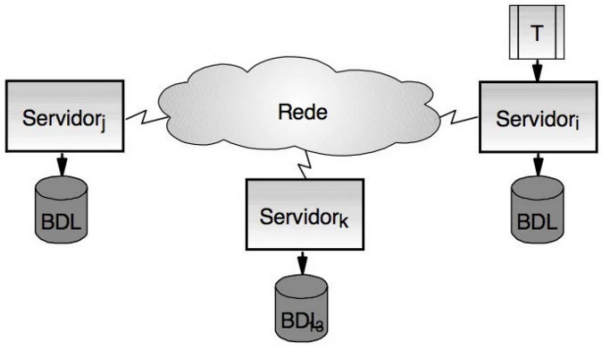


Figura 13. Base de dados distribuida

Vantagens

Os dados ficam localizados próximos aos locais de maior demanda – Os dados ficam dispersos para atender a necessidade comercial;

- Maior rapidez de acesso aos dados – Os utilizadores finais costumam trabalhar apenas com um sub-conjunto dos dados da empresa, armazenado localmente;
- Maior rapidez de processamento de dados – Um SGBDD divide a carga de trabalho do sistema, processando dados em vários locais;
- Facilidade de ampliação – É possível adicionar novos locais à rede sem afetar as operações de outros locais;
- Aprimoramento das comunicações – Como os sites locais são menores e mais próximos dos clientes, promovem melhor comunicação entre os departamentos e entre os clientes e a equipe da empresa;
- Custos operacionais reduzidos – Do ponto de vista dos custos, é mais eficiente adicionar estações de trabalho a uma rede do que actualizar um sistema de mainframe. O trabalho de desenvolvimento é feito de modo mais rápido e barato em PCs de baixo custo do que em mainframes;
- Interface amigável – Os PCs e as estações de trabalho costumam ser equipados com interface gráfica de usuário (GUI) fácil de usar. A GUI simplifica o treinamento e a utilização dos usuários finais;
- Menor risco de falha em ponto único – Quando um componente dos computadores falha, o trabalho é mantido por outras estações. Os dados também são distribuídos em vários locais;
- Independência de processador – O usuário final é capaz de acessar todas as copias disponíveis dos dados e suas solicitações são processadas por qualquer processador no local dos dados.

Desvantagens

- Complexidade de gerenciamento e controle – As aplicações devem reconhecer a localização dos dados e ter a capacidade de integrá-los a partir de vários locais. É necessário que os administradores tenham a capacidade de coordenar as atividades do BD, evitando sua degradação em função de anomalias;
- Dificuldade tecnológica – É necessário tratar e solucionar a integridade de dados, a gestão de transacções, o controle de concorrência, o backup, a recuperação, a optimização de consultas, a selecção do caminho de acesso, etc;
- Segurança – A probabilidade de falhas de segurança aumenta quando os dados são armazenados em vários locais. A responsabilidade do gerenciamento dos dados será compartilhada por diferentes pessoas em diversos locais;
- Falta de padrões – Não há protocolos de comunicação padronizado no nível de BD. (Embora o TCP/IP seja, na prática, um padrão no nível de rede, não há padronização no nível de aplicação). Por exemplo, diferentes fornecedores de BD empregam técnicas diferentes e geralmente incompatíveis de gerenciamento da distribuição de dados e processamento no ambiente de SGBDD;
- Ampliação das necessidades de armazenamento e infra-estrutura – São necessárias várias cópias de dados em diferentes locais, exigindo, assim, espaço adicional de armazenamento em disco;
- Aumento de custos com treino – Os custos com treino costumam ser mais elevados em modelos distribuídos do que em centralizados, às vezes a ponto de compensar as economias operacionais de hardware;
- Custos – Os BDD exigem uma infra-estrutura duplicada para operar (localização física, ambiente, pessoal, software, licenciamento, etc).

Cuidados com banco de dados distribuídos devem ser tomados para assegurar o seguinte:

A distribuição é transparente — Utilizadores devem poder interagir com o sistema como se ele fosse um único sistema lógico. Isso se aplica ao desempenho do sistema, métodos de acesso, entre outras coisas.

Transacções são transparentes — Cada transacção deve manter a integridade do banco de dados dentre os múltiplos bancos de dados. Transações devem também ser divididas em sub-transações, cada sub-transação afetando um sistema de banco de dados.

Conclusão

Uma Base de dados distribuída (BDD) consiste numa colecção de uma ou mais base de dados logicamente integradas e distribuídas por vários computadores (Nós) ligados por uma rede física (LAN ou WAN).

Apesar de ela se encontrar distribuída numa rede de computadores, esta surge para os utilizadores como se se tratasse de uma BD centralizada. Este é o papel do sistema gestor de BDD (SGBDD).

Avaliação da actividade

Verifique a sua compreensão!

1 Defina BDD?

2 Liste 3 vantagens e desvantagens da BDD?

3 Em que consiste a fragmentação e réplica em BDD?.

Actividades de Aprendizagem

Actividade 3.2: Conceitos Data Warehouse

Contextualização

O Data Warehouse (DW) é um conjunto de técnicas que, aplicadas em conjunto, geram um sistema de dados que proporciona informações para tomada de decisões. Ele funciona tipicamente na arquitectura cliente/servidor. Segundo Inmon (1993), considerado um pioneiro no tema, um data warehouse é uma colecção de dados orientada por assuntos, integrado, variante no tempo, e não volátil, que tem por objectivo dar suporte aos processos de tomada de decisão.

O data warehouse é uma Base de dados contendo dados extraídos do ambiente de produção da empresa, que foram seleccionados e depurados, tendo sido optimizados para processamento de consulta e não para processamento de transacções. Em comparação com os Base de dados transaccionais, o data warehouse possui enormes quantidades de dados de múltiplas fontes, as quais podem incluir Base de dados de diferentes modelos e algumas vezes arquivos adquiridos de sistemas independentes e plataformas (Elmasri e Navathe, 1994).

Em termos simples, um Data Warehouse, ou em português, Armazém de Dados, pode ser definido como um Base de dados especializado, o qual integra e gerência o fluxo de

informações a partir da Base de dados corporativos e fontes de dados externas à empresa.

Características do Data Warehouse

Segundo Inmon (1996), um DW deve ser orientado por assuntos, integrado, variável no tempo e não volátil. Essas seriam as principais características de um DW, porém existem outras também importantes como a localização, credibilidade dos dados e granularidade (Cazella, 2005).

Orientação por Assunto : A orientação por assunto é uma característica marcante de um DW, pois toda modelagem é voltada em torno dos principais assuntos da empresa. Enquanto todos os sistemas transacionais estão voltados para processos e aplicações específicas, os DWs objectivam assuntos. Os assuntos são o conjunto de informações relativas a determinada área estratégica de uma empresa. Um exemplo típico pode ser ilustrado da área de vendas como produtos, revendedores, contas e clientes.

Integração: Esta característica talvez seja a mais importante do DW. É através dela que se padroniza uma representação única para os dados de todos os sistemas que formarão a base de dados do DW. Por isso, grande parte do trabalho na construção de um DW está na análise dos sistemas transaccionais e dos dados que eles contêm. Esses dados geralmente encontram-se armazenados em vários padrões de codificação, isso se deve aos inúmeros sistemas existentes nas empresas, e que eles tenham sido codificados por diferentes analistas. Isso quer dizer que os mesmos dados podem estar em formatos diferentes.

Um exemplo clássico é o que se refere aos géneros masculino e feminino. Num sistema

OLTP, o analista convencionou que o sexo seria 1 para masculino e 0 para feminino, já em outro sistema outro analista usou para guardar a mesma informação a seguinte definição, M para masculino e F para feminino, e por fim outro programador achou melhor colocar H para masculino e M para feminino.

Como podemos ver, são as mesmas informações, mas estão em formatos diferentes, e isso num DW jamais poderá acontecer. Portanto é por isso que deverá existir uma integração de dados, convencionando-se uma maneira uniforme de armazenar os mesmos.

Variação no Tempo: Segundo Inmon (1996), os Data Warehouses são variáveis em relação ao tempo, isso nada mais é do que manter o histórico dos dados durante um período de tempo muito superior ao dos sistemas transaccionais. Diz respeito ao facto do dado em um data warehouse referir-se a algum momento específico, significando que não é actualizável, a cada ocorrência de uma mudança, uma nova entrada é criada, para marcar esta mudança.

Já que no Data Warehouse, o principal objectivo é analisar o comportamento das mesmas durante um período de tempo maior e fundamentados nessa variação é que os gerentes tomam as decisões, então num DW é frequente manter-se um horizonte de tempo bem superior ao dos sistemas transaccionais.

Deste modo é válido dizer que os dados nos sistemas transaccionais estão sendo actualizados constantemente, cuja exactidão é válida somente para o momento de acesso. Já os dados existentes num DW são como fotografias que reflectem os mesmos num determinado momento do tempo. Essas fotografias são chamadas de snapshots (Cazella, 2005).

A dimensão tempo, sempre estará presente em qualquer facto de um DW, isso ocorre porque, como citado anteriormente, sempre os dados reflectirão um determinado momento de tempo, e obrigatoriamente deverá conter uma chave de tempo para expressar a data em que os dados foram extraídos. Portanto pode-se dizer que os dados armazenados correctamente no DW não serão facilmente actualizados tendo-se assim uma imagem fiel da época em que foram gerados.

Assim como os dados, é importante frisar que os metadados, também possuem elementos temporais, porque mantêm um histórico das mudanças nas regras de negócio da empresa. Os metadados são responsáveis pelas informações referentes ao caminho do dado dentro do DW.

Não Volatilidade: No DW existem somente duas operações, a carga inicial e as consultas dos front-ends aos dados. Após serem integrados e transformados, os dados são carregados em bloco para o data warehouse, para que estejam disponíveis aos utilizadores para acesso. Isso pode ser afirmado porque a maneira como os dados são carregados e tratados é completamente diferente dos sistemas transaccionais.

No ambiente operacional, ao contrário, os dados são, em geral, actualizados registo a registo, em múltiplas transacções. Esta volatilidade requer um trabalho considerável para assegurar integridade e consistência. Um data warehouse não requer este grau de controle típico dos sistemas orientados a transacções, pois, no DW o que acontece é somente ler os dados na origem e gravá-los no destino, ou seja, no banco modelado multidimensional.

Deve-se considerar que os dados sempre passam por filtros antes de serem inseridos no DW. Com isso muitos deles jamais saem do ambiente transaccional, e outros são tão resumidos que não se encontram fora do DW. Em outras palavras, a maior parte dos dados é física e radicalmente alterada quando passam a fazer parte do DW. Do ponto de vista de integração, não são mais os mesmos dados do ambiente operacional. À luz destes factores, a redundância de dados entre os dois ambientes raramente ocorre, resultando em menos de 1 por cento de duplicações, essa definição é dada por Inmon (1996), e é muito válida.

Localização: Os dados podem estar fisicamente armazenados de três formas:

Num único local centralizado, com Base de dados em um DW integrado procura-se nessa forma maximizar o poder de processamento e agilizando a busca dos dados. Esse tipo de armazenagem é bastante utilizada, porém há o inconveniente do investimento em hardware para comportar a base de dados muito volumosa e o poderio de processamento elevado para atender satisfatoriamente as consultas simultâneas de muitos utilizadores.

Os distribuídos, em forma de Data Marts, são armazenados por áreas de interesse. Por exemplo, os dados da gerência financeira num servidor, dados de marketing noutro e dados da contabilidade num terceiro lugar. Tem por benefício bastante performance, pois não sobrecarrega um único servidor e as consultas serão sempre atendidas em tempo satisfatório.

Armazenados por níveis de detalhes, em que as unidades de dados são mantidas no DW. Pode-se armazenar dados altamente resumidos num servidor, dados resumidos com nível de detalhe intermediário no segundo servidor e os dados mais detalhados (atômicos), num terceiro servidor. Os servidores da primeira camada podem ser optimizados para suportar um grande número de acessos e um baixo volume de dados, enquanto alguns servidores nas outras camadas podem ser adequados para processar grandes volumes de dados, mas baixo número de acesso.

Credibilidade dos Dados: A credibilidade dos dados é o muito importante para o sucesso de qualquer projecto. Discrepâncias simples de todo tipo podem causar sérios problemas quando se quer extrair dados para suportar decisões estratégicas para o negócio das empresas. Dados não dignos de confiança podem resultar em relatório inúteis, que não têm importância alguma. "Se você tem dados de má qualidade e os disponibiliza em um DW, o seu resultado final será um suporte à decisão de baixo nível com altos riscos para o seu negócio", afirma Robert Craig, analista do Hurwitz Group (Data Warehouse, 2005). Coisas aparentemente simples, como um CEP errado, podem não ter nenhum impacto em uma transacção de compra e venda, mas podem influir nas informações referentes a cobertura geográfica, por exemplo. "Não é apenas a escolha da ferramenta certa que influi na qualidade dos dados", afirma Richard Rist, vice-presidente Data Warehousing Institute (Data Warehouse, 2005). Segundo ele, conjuntos de colecções de dados, processos de entrada, metadados e informações sobre a origem dos dados, são importantíssimos. Outras questões como a manutenção e actualização dos dados e as diferenças entre dados para BD transaccionais e para uso em Data Warehousing também são cruciais para o sucesso dos projectos. Além das camadas do DW propriamente dito, tem-se a camada dos dados operacionais, de onde os dados mais detalhados são colectados. Antes de fazer parte do DW estes dados passam por diversos processos de transformação para fins de integração, consistência e acurácia.

Granularidade: Granularidade nada mais é do que o nível de detalhe ou de resumo dos dados existentes num DW. Quanto maior for o nível de detalhes, menor será o nível de granularidade. O nível de granularidade afecta directamente o volume de dados armazenados no DW, e ao mesmo tempo o tipo de consulta que pode ser respondida. Quando se tem um nível de granularidade muito alto o espaço em disco e o número de índices necessários, tornam-se bem menores, porém há uma correspondente diminuição da possibilidade de utilização dos dados para atender a consultas detalhadas. Com o nível de granularidade muito baixo, é possível responder a praticamente qualquer consulta, mas uma grande quantidade de recursos computacionais é necessária para responder a perguntas muito específicas. No entanto, no ambiente de DW, dificilmente um evento isolado é examinado, é mais provável que ocorra a utilização da visão de conjunto dos dados.

Os dados levemente resumidos compreendem um nível intermediário na estrutura do DW, são derivados do detalhe de baixo nível encontrado nos dados detalhados actuais. Este nível do DW é quase sempre armazenado em disco. Na passagem para este nível os dados sofrem modificações. Por exemplo, se as informações nos dados detalhados actuais são armazenadas por dia, nos dados levemente resumidos estas informações podem estar armazenadas por semanas. Neste nível o horizonte de tempo de armazenamento normalmente fica em cinco anos e após este tempo os dados sofrem um processo de envelhecimento e podem passar para um meio de armazenamento alternativo. Os dados altamente resumidos são compactos e devem ser de fácil acesso, pois fornecem informações estatísticas valiosas para os Sistemas de Informações Executivas (EIS), enquanto que nos níveis anteriores ficam as informações destinadas aos Sistemas de Apoio a Decisão (SAD), que trabalham com dados mais analíticos procurando analisar as informações de forma mais ampla. O balanceamento do nível de granularidade é um dos aspectos mais críticos no planeamento de um DW, pois na maior parte do tempo há uma grande demanda por eficiência no armazenamento e no acesso aos dados, bem como pela possibilidade de analisar dados em maior nível de detalhes. Quando uma organização possui grandes quantidades de dados no DW, faz sentido pensar em dois ou mais níveis de granularidade, na parte detalhada dos dados. Na realidade, a necessidade de existência de mais de um nível de granularidade é tão grande, que a opção do projecto que consiste em duplos níveis de granularidade deveria ser o padrão para quase todas as empresas.

O chamado nível duplo de granularidade se enquadra nos requisitos da maioria das empresas. Na primeira camada de dados ficam os dados que fluem do armazenamento operacional e são resumidos na forma de campos apropriados para a utilização de analistas e gerentes. Na segunda camada, ou nível de dados históricos, ficam todos os detalhes vindos do ambiente operacional. Como há uma verdadeira montanha de dados neste nível, faz sentido armazenar os dados em um meio alternativo como fitas magnéticas.

Com a criação de dois níveis de granularidade no nível detalhado do DW, é possível atender a todos os tipos de consultas, pois a maior parte do processamento analítico dirige-se aos dados levemente resumidos que são compactos e de fácil acesso. E para ocasiões em que um maior nível de detalhe deve ser investigado existe o nível de dados históricos. O acesso aos dados do nível histórico de granularidade é caro, incómodo e complexo, mas caso haja necessidade de alcançar esse nível de detalhe, lá estará ele.

Implantação de um Data Warehouse

O DW não é como um software, que pode ser comprado e instalado em todos os computadores da empresa em algumas horas, na realidade sua implantação exige a integração de vários produtos e processos. O Data Warehouse (DW) apresenta um processo complexo composto por vários itens como metodologias, técnicas, máquinas, Base de dados, ferramentas de front-end, extração, metadados, refinamento de dados, replicação, e principalmente, recursos humanos.

Cada elo desta corrente está sujeito a falhas que podem transformar um projecto de milhões de reais, que era tido pela empresa como a tábua de salvação para seus problemas, num grande pesadelo (Flores, 2005).

O momento mais crucial de todo processo, e o causador da maior parte dos problemas, é o da escolha das ferramentas, das Bases de dados, das consultorias, e da definição do escopo do projecto e da selecção dos indivíduos que farão parte do staff de DW. Mais do que um Base de dados de última geração é importante seleccionar com extremo critério os profissionais que farão parte de um projecto.

Depois da escolha dos profissionais responsáveis, deve-se fazer um levantamento e definir os objectos de negócios e, por conseguinte, as questões gerênciais a que respondem. Essa etapa é de extrema importância, pois é ela que irá determinar quais serão os dados a serem armazenados no Data Warehouse. Exigirá, também, uma metodologia específica e diferente dos sistemas transaccionais. Terminada esta etapa, partiremos então para a segunda fase da implantação que consiste em fazer a modelagem dimensional, ou seja, partindo dos objectos gerências levantados faz-se a modelagem da nova Base gerando os fatos e as dimensões. Logo em seguida, parte-se para a criação física do modelo, onde as especificidades de um SGBD e da ferramenta OLAP escolhidos são levadas em consideração para optimizar as futuras consultas à Base e onde se deve dar preferência aos esquemas estrela. Feito isso, o passo seguinte é a carga dos dados na DW.

Para isso, é necessário que se definam as origens dos mesmos nos sistemas legados, ou seja, identificar em quais sistemas e onde estão armazenados. Nessa etapa é imprescindível a presença de um analista de sistemas da empresa que conheça profundamente os sistemas transaccionais, pois isso facilitará muito o trabalho de localização e identificação dos dados. Seguindo o roteiro descrito por Flores (2005) a próxima etapa é fazer as rotinas de extracção dos dados. Essas rotinas podem ser desenvolvidas por programadores em qualquer linguagem de programação. Após a extracção resta dar a carga dos dados no Base dimensional criada no segundo passo. Essa carga também pode ser feita através de rotinas desenvolvidas pelos programadores.

Concluída a carga deve-se fazer uma verificação profunda da consistência dos dados, isso é muito importante, pois está-se trabalhando com informações para dar suporte a decisão e, qualquer dado errado poderá determinar o fracasso da análise do negócio em questão. Outro passo importantíssimo na elaboração é a confecção e armazenamento dos metadados.

Os metadados são os dados de controle do Data Warehouse. Para fazermos a análise dos dados temos as ferramentas OLAP, que permitem a visualização dos dados da base dimensional e sua análise de acordo com a imaginação do executivo. Com esses softwares, também denominados ferramentas de front-end, o utilizador poderá analisar as informações que estão contidas no Base multidimensional.

Além dos passos anteriores, os utilizadores devem ser treinados de forma a aprender a manipular as informações existentes seja na criação das estatísticas, seja na criação de gráficos e relatórios.

Problemas que podem existir na Implantação de um Data Warehouse

Existem diversos problemas que podem ocorrer durante o desenvolvimento de um sistema de DW. Dentre estes problemas, segundo Bar (1996, apud Data Warehouse, 2005), os mais comuns são:

1. Não envolver a alta direcção da empresa no projecto: O projecto de um DW só terá sucesso se os futuros utilizadores se envolverem directamente desde o início nas actividades, pois isto facilitará a destinação das verbas necessárias nos momentos oportunos além de direccionar os trabalhos para que os reais objectivos do DW para o negócio da empresa sejam alcançados no momento da implantação.

2. Gerar falsas expectativas com promessas que não poderão ser cumpridas: citar frases do tipo "O DW mostrará aos gerentes as melhores decisões" pode causar tanto desconfiança no projecto quanto desprezo. O DW não mostrará as melhores decisões, mas sim respostas às consultas efectuadas. Cabe aos utilizadores elaborar consultas inteligentes e analisar as respostas obtidas.

3. Carregar no DW informações somente porque elas estão disponíveis nos sistemas transaccionais: Nem todos os dados disponíveis nos sistemas operacionais da empresa são necessariamente úteis para o DW. Cabe ao arquitecto dos dados analisar, junto com os utilizadores, quais os dados que realmente contêm informações necessárias e desprezar aqueles que não fazem parte dos objectivos do DW.

4. Imaginar que o projecto do Base de dados do DW é o mesmo que o projecto de um sistema transaccional: num processo transaccional devem ser dimensionados os recursos para que se atinja uma alta velocidade de acesso e grandes facilidades na actualização de registos. Nos sistemas de apoio à decisão a realidade é totalmente outra. O objectivo destes sistemas é fornecer acessos agregados, ou seja, somas, médias, tendências, etc.

   Outra diferença entre os dois tipos de sistemas pode ser detectado no tipo de utilizadores. Nos sistemas transaccionais um programador desenvolve uma consulta que poderá ser utilizada milhares de vezes. No DW o utilizador final desenvolve suas próprias consultas que podem ser utilizadas somente uma vez.

5. Na selecção do pessoal, escolher um gerente para o DW com orientação essencialmente técnica: os sistemas de apoio à decisão são na verdade uma prestação de serviços e não um serviço de armazenamento de dados. Por isso, é fundamental que o gerente do DW seja uma pessoa voltada aos interesses dos utilizadores e principalmente que, saiba dos termos utilizados diariamente pelos altos gerentes e outros tomadores de decisões.

6. Dedicar-se ao tratamento de dados do tipo registos numéricos e string:

   Muitos poderiam imaginar que as informações que serão utilizadas em um DW seriam oriundas especificamente dos registos das bases de dados transaccionais, e que estas informações seriam apenas números ou palavras. Porém a inclusão de textos, imagens, sons e vídeos podem ser bastante úteis no momento da análise de determinadas situações da empresa e do negócio.

7. Projectar um sistema com base em um hardware que não poderá comportar o crescimento da demanda do DW: A capacidade dos servidores está em constante crescimento, porém pode ser dimensionando um ou mais equipamentos para trabalharem por um ou dois anos, mas as características destes sistemas indicam que a quantidade de informações armazenadas pode atingir até mesmo alguns terabytes. É importante que o servidor do Base de dados do DW seja fornecido por uma empresa confiável e que garanta a possibilidade de serem realizadas expansões a valores e prazos compatíveis com os de mercado.

8. Imaginar que após a implantação do DW os problemas estarão terminados: Muito esforço deve ser despendido para que um sistema de DW saia da prancheta. Porém, após a sua implantação os utilizadores começarão a criar mais e mais consultas, e estas consultas necessitarão de novos dados que resultarão em novas consultas. Desta forma, o projecto de um sistema de apoio à decisão precisa ser revisto e actualizado constantemente, não apenas com novos dados, mas também com novas tecnologias.

Ferramentas que permitem extrair informações do Data Warehouse

Mesmo sabendo que a informação sobre o perfil do cliente típico ou do produto de sucesso de uma empresa encontra-se de alguma forma entre os muitos gigabytes de dados de marketing e de vendas armazenados nas Bases de dados da empresa, ainda pode existir um longo caminho a ser percorrido até que esta informação esteja de facto disponível. A sua extracção eficaz, de modo a poder subsidiar decisões, depende da existência de ferramentas especializadas que permitam a captura de dados relevantes mais rapidamente e a sua visualização através de várias dimensões.

O termo extracção neste contexto não deve ser confundido com a extracção dos dados das fontes para posterior alimentação do data warehouse. As ferramentas não devem apenas permitir o acesso aos dados, mas também permitir análises de dados significativas, de tal maneira a transformar dados brutos em informação útil para os processos estratégicos da empresa.

O sucesso de um data warehouse pode depender da disponibilidade da ferramenta certa para as necessidades de seus utilizadores. As ferramentas mais simples são os produtos para consultas e geradores de relatórios básicos.

Estes geradores de relatório não atendem a utilizadores que precisem mais do que uma visão estática dos dados e que não pode mais ser manipulada.

Ferramentas OLAP podem oferecer a este tipo de utilizador maior capacidade de manipulação, permitindo analisar o porquê dos resultados obtidos. Estas ferramentas, muitas vezes, são baseadas em Bases de dados multidimensionais, o que significa que os dados precisam ser extraídos e carregados para as estruturas proprietárias do sistema, já que não há padrões abertos para o acesso de dados multidimensionais - outra solução oferecida por fornecedores nesta área é o OLAP relacional (ROLAP) e o OLAP multidimensional (MOLAP).

O OLAP não é uma solução imediata, configurar o programa de OLAP e ter acesso aos dados requer uma clara compreensão dos modelos de dados da empresa e das funções analíticas necessárias aos executivos e outros analistas de dados. Comparativamente ao OLAP, Sistemas de Informações Executivas (SIE) apresentam uma visualização de dados mais simplificada, altamente consolidada e, na maior parte das vezes estática. Até porque, em geral, os executivos não dispõem do tempo e da experiência para executar uma análise OLAP.

O data mining ou mineração de dados é uma categoria de ferramentas de análise open-end. Ao invés de fazerem perguntas, os utilizadores entregam para a ferramenta grandes quantidades de dados em busca de tendências ou agrupamentos dos dados. A diferença entre sistemas do tipo SIE e a tecnologia de data mining pode ser vista da seguinte forma: se você tem perguntas específicas e sabe os dados de que necessita, utilize um SIE; quando você não sabe qual a pergunta, mas mesmo assim precisa de respostas, use data mining.

Conclusão

Um Data Warehouse permite a geração de dados integrados e históricos auxiliando os directores a decidirem embaçados em factos e não em intuições ou especulações, o que reduz a probabilidade de erros, aumentados à velocidade, na hora da decisão. Esse instrumento estudado mostra o quanto ele é útil para os gerentes das empresas para a obtenção e análise da informação, se realizado da forma correcta. Também é interessante ressaltar que não é um mecanismo de fácil implementação, pois exige os mais diversos recursos e dentre eles o recurso humano é que se destaca. Umas séries de ferramentas são necessárias para que se possa desfrutar de seus benefícios e vários aspectos relacionados aos modelos dimensionais que permitem uma organização dos processos envolvidos.

Conhecer mais sobre essa tecnologia permitirá aos administradores descobrir novas maneiras de diferenciar sua empresa numa economia globalizada, deixando-os mais seguros para definirem as metas e adoptarem diferentes estratégias em sua organização, conseguindo assim visualizar, antes de seus concorrentes, novos mercados e oportunidades actuando de maneiras diferentes conforme o perfil de seus consumidores.

Avaliação da actividade

Verifique a sua compreensão!

1. Defina o conceito data warehouse?
2. Qual a vantagem da utilização do DW nas empresas?
3. Liste pelo menos dois problemas relacionados com a implantação do DW.

Actividades de Aprendizagem

Actividade 3.3: Conceitos Data Mining

Contextualização

Prospecção de dados ou mineração de dados (também conhecida pelo termo inglês data mining) é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.

No campo da administração, a mineração de dados é o uso da tecnologia da informação para descobrir regras, identificar factores e tendências-chave, descobrir padrões e relacionamentos ocultos em grandes bancos de dados para auxiliar a tomada de decisões sobre estratégia e vantagens competitivas.

Visão geral sobre Data Mining

A mineração de dados é formada por um conjunto de ferramentas e técnicas que através do uso de algorítmos de aprendizagem ou classificação baseados em redes neurais e estatística, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento.

Esse conhecimento pode ser apresentado por essas ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão, grafos, ou dendrogramas.

O ser humano sempre aprendeu observando padrões, formulando hipóteses e testando-as para descobrir regras. A novidade da era do computador é o volume enorme de dados que não podem ser examinados, à procura de padrões sem um prazo razoável. A solução é instrumentalizar o próprio computador para detectar relações que sejam novas e úteis. A mineração de dados (MD) surge para essa finalidade e pode ser aplicada tanto para a pesquisa científica como para impulsionar a lucros da empresa.

Também a multidisciplinaridade da mineração de dados pode ser considerada inevitável devido à integração de diversas áreas de conhecimento no processo de análise, abordando áreas de pesquisas que envolvem estatística, matemática e a computação, as quais são disciplinas fundamentais para realização do processo de mineração de dados.

Diariamente as empresas acumulam grande volume de dados em seus aplicativos operacionais. São dados brutos que dizem quem comprou o quê, onde, quando e em que quantidade. É a informação vital para o dia-a-dia da empresa. Se fizermos estatística ao final do dia para repor stocks e detectar tendências de compra, estaremos praticando business intelligence (BI).

Se analisarmos os dados com estatística de modo mais refinado, à procura de padrões de vinculações entre as variáveis registadas, então estaremos fazendo mineração de dados. Buscamos com a MD conhecer melhor os clientes, seus padrões de consumo e motivações.

A MD resgata em organizações grandes o papel do dono atendendo no balcão e conhecendo sua clientela. Através da MD, esses dados agora podem agregar valor às decisões da empresa, sugerir tendências, desvendar particularidades dela e de seu meio ambiente e permitir acções melhor informadas aos seus gestores.

Pode-se então diferenciar o business inteligence (BI) da mineração de dados (MD) como dois patamares distintos de actuação.

- O primeiro busca subsidiar a empresa com conhecimento novo e útil acerca do seu meio ambiente e funciona no plano estratégico.
- O segundo visa obter a partir dos dados operativos brutos, informação útil para subsidiar a tomada de decisão nos escalões médios e altos da empresa e funciona no plano táctico.

Etapas da mineração de dados

Os passos fundamentais de uma mineração bem sucedida a partir de fontes de dados (base de dados, relatórios, logs de acesso, transacções, etc.) consistem de uma limpeza (consistência, preenchimento de informações, remoção de ruído e redundâncias, etc.). Disto nascem os repositórios organizados (Data Marts e Data Warehouses).

É a partir deles que se pode seleccionar algumas colunas para atravessarem o processo de mineração. Tipicamente, este processo não é o final da história: de forma interactiva e frequentemente usando visualização gráfica, um analista refina e conduz o processo até que os padrões apareçam. Observe que todo esse processo parece indicar uma hierarquia, algo que começa em instâncias elementares (embora volumosas) e terminam em um ponto relativamente concentrado.

Encontrar padrões requer que os dados brutos sejam sistematicamente "simplificados" de forma a desconsiderar aquilo que é específico e privilegiar e/ou valorizar tudo o que for generalizado. Em um determinado produto uma única data pode apenas significar que esse cliente em particular procurava grande quantidade desse produto naquele exato momento. Mas isso provavelmente não indica nenhuma tendência de mercado.

Tipos de informação obtidos com a Mineração de Dados

Com o uso da Mineração de dados, é possível descobrir informações relacionadas a associações, sequências, classificação, aglomeração e prognósticos.

- **Associações**: São ocorrências ligadas a um único evento. Por exemplo: um estudos de modelos de compra em supermercados pode revelar que, na compra de salgadinhos de milho, compra-se também um refrigerante tipo cola em 65% das vezes: mas, quando há uma promoção, o refrigerante é comprado em 85% das vezes.Com essas informações, os gerentes podem tomar decisões mais acertadas pois aprenderam a respeito da rentabilidade de uma promoção.
- **Sequências**: Na sequência os eventos estão ligados ao longo do tempo. Pode-se descobrir, por exemplo, que quando se compra uma casa, em 65% as vezes se adquire uma nova geleira no período de duas semanas; e que em 45% das vezes, um fogão também é comprado um mês após a compra da residência.
- **Classificação**: Reconhece modelos que descrevem o grupo ao qual o item pertence por meio do exame dos itens já classificados e pela inferência de um conjunto de regras. Exemplo: empresas de operadoras de cartões de crédito e companhias telefónicas preocupam-se com a perda de clientes regulares, a classificação pode ajudar a descobrir as características de clientes que provavelmente virão abandoná-las e oferecer um modelo para ajudar os gerentes a prever quem são, de modo que se elabore antecipadamente campanhas especiais para reter esses clientes.
- **Aglomeração (clustering):** Funciona de maneira semelhante à classificação quando ainda não foram definidos grupos. Uma ferramenta de data mining descobrirá diferentes agrupamentos dentro da massa de dados. Por exemplo ao encontrar grupos de afinidades para cartões bancários ou ao dividir o banco de dados em categorias de clientes com base na demografia e em investimentos pessoais.

Embora todas essas aplicações envolvam previsões, os prognósticos as utilizam de modo diferente. Partem de uma série de valores existentes para prever quais serão os outros valores. Por exemplo um prognóstico pode descobrir padrões nos dados que ajudam os gerentes a estimar o valor futuro de variáveis com números de vendas.

Esses sistemas realizam uma análise de alto nível quanto a padrões ou tendências, mas também podem esmiuçar os dados para revelar mais detalhes, se necessário. Existem aplicações de data mining para todas as áreas funcionais da empresa, bem como para o trabalho científico ou governamental. É como usar o data mining para analisar, detalhadamente, padrões em dados sobre consumidores e, a partir disso, montar campanhas de marketing um-a-um ou identificar clientes lucrativos. (LAUDON & LAUDON, 2011, p.159).

Conclusão

O conceito data mining ainda é um desafio para a maior parte das instituições, pois a prospecção de dados ou mineração de explorar grandes quantidades de dados e as instituições ainda estão no processo de ter os seus dados armazenados em base de dados de forma lógica. Portanto após concluir esta actividade explore mais este conceito pesquisando na Internet e lendo mais livros relacionados com data mining.

Avaliação da actividade

Verifique a sua compreensão!

1. Em que consiste data mining?
2. O que podemos obter com o data mining?
3. Relacione data werehouse e data mining.

Resumo da Unidade 3

Nesta unidade abordamos o conceito de Base de Dados Distribuida, Dataware House e Data Mining. No final de cada Unidade, o estudante deve procurar responder as questões de avaliação, por forma a medir o seu nível de compreensão.

Aconselhamos também que faça resumos sobre cada uma das actividades aqui apresentadas e pesquise sobre as mesmas por forma a ter uma visão mais ampla sobre a os termos aqui tratados.

Avaliação da Unidade 3

Verifique a sua compreensão!

1. Defina Base de Dados Distribuída?

   Sol: Uma Base de dados distribuída consiste numa colecção de uma ou mais base de dados logicamente integradas e distribuídas por vários computadores ligados por uma rede física

2. Quando é que podemos considerar uma replica síncrona?

   Sol: Numa replicação síncrona, cada transacção é dada como concluída quando todos os nós confirmam que a transacção local foi bem sucedida

3. Em que consiste o data mining?

   Sol: data mining é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.

4. Em que consiste o data warehouse?

   Sol:O Data Warehouse é um conjunto de técnicas que aplicadas em conjunto geram um sistema de dados que proporcionam informações para tomada de decisões.

Leituras e outros Recursos

- http://www.diegomacedo.com.br/sistema-de-gerenciamento-de-banco-de-dados-distribuidos-sgbdd/ acedido via Internet no dia 10.04.2016
- https://pt.wikipedia.org/wiki/mineracação_de_dados acedido via Internet no dia 10.04.2016
- Tecnologia de Bases de Dados (3ª Edição), José Luís Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431

Unidade 4. Segurança de base de dados

Introdução à Unidade

A forma como os SGBD lidam com a segurança de dados e o acesso às informações é realizada a partir de um conjunto de permissões. A atribuição de permissões é normalmente realizada através de comandos GRANT. Para eliminar uma permissão ou privilegio, utiliza-se o comando REVOKE.

A forma como a generalidade dos sistemas lida com estas questões é diferente, sendo normalmente disponibilizado um utilitário que permite fazer a distribuição mais ou menos apoiada e de forma mais ou menos visual. Por exemplo o Access faz a gestão da segurança através de Menus da sua aplicação.

A segurança em Banco de Dados é constituída pelo estudo do conjunto de medidas, por mecanismos e políticas de segurança que se destinam a prover dispositivos, a fim de que sejam mantidos os aspectos de disponibilidade, de confidencialidade e integridade dos dados, além de fornecer protecção aos sistemas contra possíveis acidentes, falhas ou ataques perpetrados interna ou externamente às empresas.

Tais medidas de segurança têm como um dos principais mecanismos de continuidade os procedimentos de backup e recovery, sendo o primeiro o acto de realizar cópias de segurança e o segundo a forma de recuperar tais cópias, o que veremos durante o estudo desta unidade. Portanto a abordagem deste assunto será generalizada, pois são faremos menção a aplicações ou aplicativos específicos.

Objectivos da Unidade

Após a conclusão desta unidade, deverás ser capaz de:

1. Atribuir e remover privilégios numa BD;
2. Aplicar os conceitos de backup e recovery;
3. Distinguir o conceito backup e recovery;
4. Diferenciar os tipos de backup.

Actividades de Aprendizagem

Actividade 4.1. Tipo de Segurança

Contextualização

A segurança em base de dados é uma área bastante ampla que se refere a muitas questões, incluindo Questões legais e éticas referentes ao direito de acesso a certas informações. Algumas informações podem ser consideradas privadas e não podem ser acedidas legalmente por pessoas não autorizadas.

Questões políticas nos níveis governamental, institucional ou corporativo, como quais os tipos de informação que não devem ser tornadas disponíveis publicamente — por exemplo, avaliações de crédito e registos médicos pessoais.

Questões relacionadas ao sistema, como os níveis de sistema nos quais as várias funções de segurança devem ser implementadas; por exemplo, se uma função de segurança deve ser tratada no nível físico de hardware, no nível de sistema operacional ou no nível do SGBD.

As necessidades em algumas organizações de identificar múltiplos níveis de segurança e em categorizar os dados e utilizadores com base nessas classificações — por exemplo, altamente secreto (top secret), secreto (secret), confidencial (confidential) e não confidencial (unclassified). A política de segurança de uma organização a respeito da permissão de acesso a várias classificações de dados deve ser implementada.

Ameaças as Base de Dados

As ameaças as base de dados resultam na perda ou na degradação de alguns ou de todos os seguintes objectivos de segurança: integridade, disponibilidade e confidencialidade.

Perda de integridade: A integridade da base de dados refere-se à exigência de que a informação seja protegida contra a modificação imprópria.

A modificação de dados inclui a criação, a inclusão, a alteração, a mudança de status do dado e a exclusão. A integridade é perdida se modificações não autorizadas são realizadas nos dados tanto por actos intencionais quanto por actos acidentais. Se a perda de integridade do sistema ou dos dados não for corrigida, o uso prolongado do sistema contaminado ou de dados corrompidos pode resultar em imprecisão, fraude ou em decisões equivocadas.

Perda de disponibilidade: A disponibilidade de base de dados refere-se a tornar os objectos disponíveis para um utilizador humano ou para um programa que tenham direito legítimo a eles.

Perda de confidencialidade: A confidencialidade de uma base de dados refere-se à protecção dos dados contra a divulgação não autorizada. O impacto da divulgação não autorizada de informação confidencial pode variar desde a violação da Data Privacy Act (Lei de Privacidade de Dados, nos Estados Unidos) até a colocação em risco da segurança nacional. A divulgação não autorizada, imprevista, ou não intencional pode resultar na perda da confiança pública, em constrangimentos, ou na acção legal contra a organização.

Para proteger a base de dados contra esses tipos de ameaças, quatro tipos de medidas podem ser implementadas: Controle de acesso, controle de inferência, controle de fluxo e criptografia.

Em um sistema de base de dados multiutilizadores, o SGBD deve oferecer técnicas que habilitam certos utilizadores ou grupos de utilizadores a aceder a partes seleccionadas de uma base de dados sem obter acesso ao restante da base de dados. Isso é especialmente importante quando uma grande base de dados integrados é utilizada por muitos utilizadores diferentes dentro de uma mesma organização. Por exemplo, informações sensíveis, como salários de funcionários ou avaliações de desempenho, devem ser mantidas confidencialmente em relação à maioria dos utilizadores do sistema de base de dados.

Tipicamente um SGBD inclui um subsistema de segurança de base de dados e de autorização que é responsável por garantir a segurança de partes de uma BD contra o acesso não autorizado.

Actualmente é comum se referir a dois tipos de mecanismos de segurança de BD:

Mecanismos de acesso discricionário: São utilizados para conceder privilégios a utilizadores, inclusive a capacidade de aceder a arquivos de dados, registos ou campos específicos de uma maneira especifica (como leitura, inclusão, exclusão ou actualização).

Mecanismos de acesso obrigatório (mandatory): São utilizados para impor a segurança em vários níveis por meio da classificação dos dados e dos utilizadores em várias classes de segurança (ou níveis) e, depois, pela implementação da política de segurança adequada da organização. Por exemplo, uma política de segurança típica é permitir aos utilizadores em determinado nível da classificação verem apenas os itens de dados classificados no próprio nível de classificação do utilizador (ou abaixo).

Uma extensão disso é a segurança baseada em papéis (role-based), que impõe políticas e privilégios baseando-se no conceito de papéis.

Um segundo problema comum a todos os sistemas de computação é prevenir que pessoas não autorizadas acedam ao próprio sistema, seja para obter informação, seja para realizar alterações mal-intencionadas em uma parte da BD.

O Mecanismo de segurança de uma BD deve incluir providências para a restrição de acesso ao sistema da base de dados como um todo. Essa função é chamada controle de acesso e é tratada por meio da criação de contas de utilizadores e senhas para controlar o processo de login pelo SGBD.

Um terceiro problema de segurança associado a base de dados é o de controlar o acesso a uma BD estatística, o qual é utilizado para prover informações estatísticas ou resumos de valores baseados em vários critérios. Por exemplo, uma base de dados de estatísticas de população pode fornecer informações baseadas em grupos de idade, níveis de renda, tamanhos de residência, níveis de educação e em outros critérios.

Utilizadores de base de dados estatísticas, como estatísticos do governo ou de empresas de pesquisas de mercado, têm a permissão de acesso a base de dados para recuperar informações estatísticas sobre uma população, porém, não a têm para aceder as informações confidenciais detalhadas de indivíduos em particular.

A segurança em BD estatísticas deve assegurar que informações individuais não possam ser acedidas. Às vezes, é possível deduzir ou inferir certos factos a respeito de indivíduos a partir de consultas que envolvam apenas estatísticas sumárias de grupos; consequentemente, isso também não pode ser permitido. As medidas correspondentes são chamadas controle de inferência.

Uma outra questão de segurança é a do controle de fluxo, que previne que a informação flua de tal maneira que chegue a utilizadores não autorizados. Canais que são caminhos para a informação fluir implicitamente em maneiras que violam a política de segurança de uma organização são chamados canais secretos (covert channels).

Uma última questão de segurança é a criptografia de dados, utilizada para proteger dados sensíveis (como números de cartões de crédito) que estão sendo transmitidos por meio de alguma rede de comunicação.

A criptografia também pode ser usada para prover protecção adicional para as partes sensíveis de uma base de dados. Os dados são codificados utilizando algum algorítmo de codificação. Um utilizador não autorizado que acesse os dados codificados terá dificuldades em decifrá-los; já aos utilizadores autorizados são fornecidos algorítmos de descodificação ou de decifragem (ou chaves) para esses dados. Técnicas de criptografia que são muito difíceis de serem descodificadas sem uma chave têm sido desenvolvidas para aplicações militares.

Uma discussão completa sobre segurança em sistemas de computação e base de dados está fora do escopo deste módulo. Aqui, damos apenas uma breve visão geral das técnicas de segurança em base de dados. O leitor interessado pode consultar as diversas referências discutidas na bibliografia seleccionada no final desta unidade para uma abordagem mais abrangente.

Segurança de Base de Dados e o DBA

O administrador de base de dados (DBA) é a autoridade principal para a gestão de um sistema de BD. As responsabilidades do DBA incluem a concessão de privilégios a utilizadores que precisam utilizar o sistema e a classificação de utilizadores e dados de acordo com a política da organização. O DBA possui uma conta de DBA no SGBD, às vezes chamada conta de sistema ou de super-utilizador, que habilita capacidades que não estão disponíveis para as contas e utilizadores comuns da BD. Os comandos de privilégio do DBA incluem comandos para conceder e revogar privilégios para contas individuais de utilizadores ou de grupos de utilizadores, e comandos para a realização dos seguintes tipos de acções:

1. Criação de contas: Cria uma nova conta e senha para um utilizador ou um grupo de utilizadores para habilitar o acesso ao SGBD.
2. Concessão de privilégio: Permite ao DBA conceder certos privilégios a certas contas.
3. Revogação de privilégios: Permite que o DBA revogue (cancele) certos privilégios que foram concedidos anteriormente a certas contas.
4. Atribuição de nível de segurança: Consiste em atribuir as contas de utilizadores ao nível de classificação de segurança adequado.

O DBA é responsável pela segurança geral do sistema de base de dados. A acção 1 na lista anterior é utilizada para controlar o acesso ao SGBD como um todo, ao passo que as acções 2 e 3 são usadas para controlar a autorização discricionária na base de dados, e a acção 4 é utilizada para controlar a autorização obrigatória.

Segurança ao nível do Utilizador

A forma mais básica de segurança consiste em limitar o acesso dos utilizadores aos dados através de um conjunto de Login/Password.

Alguns SGBD permitem a criação e manutenção de utilizadores através de comandos da linguagem SQL, por vezes variando a forma de comando.

CREATE USER **nome_de_usuario** IDENTIFIED BY **sua_senha** [DEFAULT TABLESPACE **nome_da_tablespace]** [TEMPORARY TABLESPACE **tablespace_temporaria**];

Exemplos:

- nome_de_usuário – É nome do utilizador que será criado;
- sua_senha – É a senha para o utilizador que está sendo criado;
- nome_da_tablespace – É a tablespace padrão onde os objectos do banco de dados são armazenados. Se essa opção for omitida, o banco assume a tablespace SYSTEM padrão;

- tablespace_temporária – É a tablespace padrão onde são armazenados os objectos temporários, como tabelas temporárias por exemplo. Se essa opção for omitida um tablaspace temporário TEMP é assumida.

CREATE USER Devan **identified by** 123;

ALTER USER Devan **identified by** abc;

DROP USER Devan; ---------- {sintaxe ORACLE}

DELETE FROM User **Where** User = 'Devan';------ {sintaxe MySQL}

Privilégios de Sistema

São os privilégios que permitem executar instruções DDL, tais como create session,create sequence, create synonym,create table,create view dentre vários outros.

Atribuir Privilégios de Sistema

O comando GRANT possui duas funcionalidades básicas: conceder privilégios para um objecto da BD (tabela, visão, sequência, banco de dados, função, linguagem procedural, esquema e espaço de tabelas), e conceder o privilégio de ser membro de um papel. Estas duas funcionalidades são semelhantes em muitos aspectos, mas são suficientemente diferentes para serem descritas em separado.Exemplo:

GRANT previlegios(s) **ON** objecto **TO** utilizador (es) [**WITH GRANT OPTION**]

GRANT insert, delete **ON** FUNCIONARIO **TO** Devan [**WITH GRANT OPTION**]

Nota: a opção WITH GRANT OPTION, o receptor do previlégio fica com a possibilidade de o atribuir a terceiros.

Remover Privilégios de Sistema

Quando alguém que não é o dono do objecto tenta conceder privilégios para o objecto, o comando falha inteiramente caso o utilizador não possua ao menos um privilégio para o objecto. Se o utilizador possuir algum privilégio para o objecto o comando prosseguirá, mas só concederá os privilégios para os quais o utilizador tem a opção de concessão.

A forma GRANT ALL PRIVILEGES emite uma mensagem de advertência quando o utilizador não possui ao menos uma opção de concessão, enquanto as outras formas emitem uma mensagem de advertência quando o utilizador não possui opção de concessão para algum dos privilégios especificamente identificados no comando (Em princípio estas informações também se aplicam ao dono do objecto, mas como o dono é sempre tratado como possuindo todas as opções de concessão estes casos nunca ocorrem).

O comando REVOKE é utilizado para revogar os privilégios de acesso.

```
REVOKE [ GRANT OPTION FOR ] privilégios
  ON objeto [ ( coluna [, ...] ) ]
  FROM { PUBLIC | nome_do_usuário [, ...] }
  { RESTRICT | CASCADE }
```

Exemplo:

REVOKE insert, delete **ON** FUNCIONARIO TO Devan [**WITH GRANT OPTION**]

OU **REVOKE** all **ON** FUNCIONARIO **TO** Devan;

Criando e Restaurando Backup via linha de comando

Criando Backup: Este comando irá criar um backup do banco em um arquivo compactado com o formato Gzip.

Digite:

```
mysqldump –u USUARIO –p SENHA BANCODEDADOS | gzip >
NOMEDOARQUIVODEBACKUP.gz
```

Restaurando o Backup: Estes dois comandos irão descompactar o arquivo GZip, e restaurá-lo.

Digite:

```
gunzip NOMEDOARQUIVODEBACKUP.gz
mysql –u USUARIO –p SENHA BANCODEDADOS < NOMEDOARQUIVO.sql
```

Conclusão

A segurança de Base de dados é extremamente importante para qualquer instituição que faz uso de sistemas de informação. Nesta actividade, abordamos o conceito de segurança num sentido mais amplo, tentando trazer os conceitos fundamentais para este tópico, tais como: Tipo de segurança, ameaças da Base de dados e Níveis de Segurança. Nas actividades que se seguem continuaremos a abordar alguns destes conceitos de forma isolada.

Avaliação da actividade

Verifique a sua compreensão!

- Por que razão devemos nos preocupar com a segurança das Bases de Dados nas nossas Instituições?
- Quais são as principais ameaças de uma BD?
- Liste pelo menos quatro competências de um DBA.
- Qual é a diferença entre os comandos GRANT e REVOKE?

Actividades de Aprendizagem

Actividade 4.2. Controlo de Acesso à Base de dados

Contextualização

O método mais simples de controlar o acesso à informação é permitir, unicamente esse acesso a partir de um conjunto de login/password que o SGBD irá validar.

Estamos assim, a partir de um pressuposto que, a partir de um determinado momento, todas as acções realizadas sobre a BD são identificadas (de alguma forma) pelo utilizador que as executa, permitindo verificar se o perfil do utilizador está ou não de acordo com as acções que este pretende levar a feito.

Por outro lado, embora um utilizador possa ter acesso a BD, os seus movimentos dentro da BD poderão ter de ser limitados.

Também podem se usar view para permitir a restrição de acesso a algumas informações presentes em tabelas. No entanto, para além de restringir o acesso a informação já existente, poderemos ter a necessidade de restringir também o conjunto de acções que cada utilizador pode realizar sobre a informação a que pode ter acesso.

Talvez não seja má ideia restringir o conjunto de utilizadores que poderão executar o comando DROP DATABASE.

Controle de acesso discricionário baseado na concessão e na revogação de privilégios

O método típico de imposição de controle de acesso discricionário em um sistema de BD é baseado na concessão e na revogação de privilégios. Muitos SGBDs relacionais actuais usam alguma variação dessa técnica. A principal ideia é incluir sentenças na linguagem de consulta que permitam ao DBA e aos utilizadores seleccionados concederem e revogarem privilégios.

Tipos de Privilégios Discricionários

O conceito de um identificador de autorização é usado para, grosso modo, referenciar uma conta de utilizador (ou grupo de contas de utilizadores). Para simplificar, usaremos as palavras utilizador ou conta indistintamente no lugar de identificador de autorização. O SGBD deve oferecer acesso selectivo a cada relação da BD baseando-se em contas específicas. As operações também podem ser controladas; assim, possuir uma conta não necessariamente habilita o possuidor a todas as funcionalidades oferecidas pelo SGBD.

Informalmente existem dois níveis para a atribuição de privilégios para o uso do sistema de base de dados:

- O nível de conta: Nesse nível, o DBA estabelece os privilégios específicos que cada conta tem, independentemente das relações na BD.
- O nível de relação (ou tabela): Nesse nível, o DBA pode controlar o privilégio para aceder cada relação ou visão individual na BD.

A generalidade dos SGBD implementa os mecanismos de segurança a três níves distintos:

- Utilizador;
- Papel que o utilizador desempenha no sistema;
- Privilégios/Permissões.

Segurança ao nível do Utilizador

A forma mais básica de segurança consiste em limitar o acesso dos utilizadores aos dados através de um conjunto de Login/Password.

Alguns SGBD permitem a criação e manutenção de utilizadores através de comandos da linguagem SQL, por vezes variando a forma de comando.

Exemplos:

CREATE USER **nome_de**_usuario IDENTIFIED BY **sua_senha** [DEFAULT TABLESPACE **nome_da_tablespace**] [TEMPORARY TABLESPACE t**ablespace_temporaria]**;

- nome_de_usuario – É nome do utilizador que será criado;
- sua_senha – É a senha para o utilizador que está sendo criado;
- nome_da_tablespace – É a tablespace padrão onde os objectos do banco de dados são armazenados. Se essa opção for omitida, o banco assume a tablespace SYSTEM padrão;
- tablespace_temporária – É a tablespace padrão onde são armazenados os objectos temporários, como tabelas temporárias por exemplo. Se essa opção for omitida um tablaspace temporário TEMP é assumida.

CREATE USER Devan **identified by** 123;

ALTER USER Devan **identified by** abc;

DROP USER Devan; ---------- {sintaxe ORACLE}

DELETE FROM User **Where** User = 'Devan';------ {sintaxe MySQL}

Privilégios de Sistema

São os privilégios que permitem executar instruções DDL, tais como create session,create sequence, create synonym,create table,create view dentre vários outros.

Atribuir Privilégios de Sistema

O comando GRANT possui duas funcionalidades básicas: conceder privilégios para um objecto da BD (tabela, visão, sequencia, banco de dados, função, linguagem procedural, esquema e espaço de tabelas), e conceder o privilégio de ser membro de um papel. Estas duas funcionalidades são semelhantes em muitos aspectos, mas são suficientemente diferentes para serem descritas em separado.

GRANT previlegios(s) **ON** objecto **TO** utilizador (es) [**WITH GRANT OPTION**]

Exemplo: **GRANT** insert, delete **ON** FUNCIONARIO **TO** Devan [**WITH GRANT OPTION**]

Nota: Com a opção WITH GRANT OPTION, o receptor do privilégio fica com a possibilidade de o atribuir a terceiros.

Remover Privilégios de Sistema

Quando alguém que não é o dono do objecto tenta conceder privilégios para o objecto, o comando falha inteiramente, caso o utilizador não possua ao menos um privilégio para o objecto. Se o utilizador possuir algum privilégio para o objecto o comando prosseguirá, mas só concederá os privilégios para os quais o utilizador tem a opção de concessão.

A forma GRANT ALL PRIVILEGES emite uma mensagem de advertência quando o utilizador não possui ao menos uma opção de concessão, enquanto as outras formas emitem uma mensagem de advertência quando o utilizador não possui opção de concessão para algum dos privilégios especificamente identificados no comando (Em princípio estas informações também se aplicam ao dono do objecto, mas como o dono é sempre tratado como possuindo todas as opções de concessão estes casos nunca ocorrem).

O comando REVOKE é utilizado para revogar os privilégios de acesso.

```
REVOKE [ GRANT OPTION FOR ] privilégios
  ON objeto [ ( coluna [, ...] ) ]
  FROM { PUBLIC | nome_do_usuário [, ...] }
  { RESTRICT | CASCADE }
```

Exemplo: **REVOKE** insert, delete **ON** FUNCIONARIO TO Devan [**WITH GRANT OPTION**]

OU REVOKE all **ON** FUNCIONARIO **TO** Devan;

Criando e Restaurando Backup via linha de comando

Criando Backup: Este comando irá criar um backup do banco em um arquivo compactado com o formato GZip

Digite:

```
mysqldump –u USUARIO –p SENHA BANCODEDADOS | gzip >
NOMEDOARQUIVODEBACKUP.gz
```

Restaurando o Backup: Estes dois comandos irão descompactar o arquivo GZip, e restaurá-lo.

Digite:

```
gunzip NOMEDOARQUIVODEBACKUP.gz
mysql –u USUARIO –p SENHA BANCODEDADOS < NOMEDOARQUIVO.sql
```

Conclusão

O controlo de acesso a base de dados é fundamental para podermos gerir os nossos utilizadores, podendo ser feito a vários níveis, desde o controle de acesso discricionário baseado na concessão e na revogação de privilégios até ao privilégio de Sistema. Portanto, a generalidade dos SGBD implementa os mecanismos de segurança ao nível do utilizador, o papel que o utilizador desempenha no sistema e os privilégios/permissões.

Avaliação da actividade

Verifique a sua compreensão!

1. O que entende por controle de acesso discricionário?
2. Em que consiste a segurança ao nível do utilizador?
3. Qual é a necessidade de utilizarmos o GRANT ALL PRIVILEGES ?

Actividades de Aprendizagem

Actividade 4.3. Backup

Contextualização

O backup refere-se à técnica de realização de cópia de segurança dos dados de um dispositivo de armazenamento para outro com o objetivo de que posteriormente, esses dados possam ser recuperados (recovery) no caso de algum problema ou por uma necessidade específica.

A origem da palavra backup vem da chamada "era industrial" e foi utilizada para identificar as peças de reposição dos carros que saiam das linhas de montagem e das máquinas e equipamentos sobressalentes que as substituíam, em caso de defeito. Assim, da mesma forma, realizar um backup é fazer uma cópia de segurança de determinado trabalho, arquivo, base de dados ou sistema, a fim de que, no caso de uma eventualidade que impeça o seu uso ou leitura, estes possam ser rapidamente recuperados, minimizando, assim, os prejuízos e permitindo a continuidade dos serviços.

Portanto, de uma forma geral, o backup corresponde a uma tarefa essencial para todos os que utilizam computadores e outros dispositivos informatizados. Actualmente, os mais conhecidos dispositivos de gravação para os backups são: as fitas magnéticas ou DAT, os CD-ROMs, os DVDs, o Blue-Ray e os discos rígidos externos. Além destes, existem inúmeros softwares que podem ser utilizados para criação de backups e para posterior recuperação dos dados (recovery).

Basicamente, a importância dos backups é assegurar a integridade dos dados contra possíveis problemas que possam eventualmente ocorrer nos sistemas, mais equipamentos ou nas unidades de armazenamento utilizadas. Além disso, devem assegurar a recuperação de arquivos dos utilizadores que possam ser apagados ou corrompidos acidentalmente ou algum tipo de vírus ou outras pragas virtuais.

Backups Completos

O backup completo é simplesmente fazer a cópia de todos os arquivos para o diretório de destino (ou para os dispositivos de backup correspondentes), independente de versões anteriores ou de alterações nos arquivos desde o último backup. Este tipo de backup é o tradicional e a primeira idéia que vêm à mente das pessoas quando pensam em backup: guardar TODAS as informações.

Outra característica do backup completo é que ele é o ponto de início dos outros métodos citados abaixo. Todos usam este backup para assinalar as alterações que deverão ser salvas em cada um dos métodos.

- A vantagem dessa solução é a facilidade para localizar arquivos que porventura devam ser restaurados.
- A grande desvantagem dessa abordagem é que leva-se muito tempo fazendo a cópia de arquivos, quando poucos destes foram efetivamente alterados desde o último backup.

Este tipo consiste no backup de todos os arquivos para a mídia de backup. Conforme mencionado anteriormente, se os dados sendo copiados nunca mudam, cada backup completo será igual aos outros. Esta similaridade ocorre devido o facto que um backup completo não verificar se o arquivo foi alterado desde o último backup; copia tudo indiscriminadamente para a mídia de backup, tendo modificações ou não. Esta é a razão pela qual os backups completos não são feitos o tempo todo pois todos os arquivos seriam gravados na mídia de backup.

Isto significa que uma grande parte da mídia de backup é usada mesmo que nada tenha sido alterado. Fazer backup de 100 gigabytes de dados todas as noites quando talvez 10 gigabytes de dados foram alterados não é uma boa prática; por este motivo os backups incrementais foram criados.

Backups Incrementais

Ao contrário dos backups completos, os backups incrementais primeiro verificam se o horário de alteração de um arquivo é mais recente que o horário de seu último backup. Se não for, o arquivo não foi modificado desde o último backup e pode ser ignorado desta vez. Por outro lado, se a data de modificação é mais recente que a data do último backup, o arquivo foi modificado e deve ter seu backup feito. Os backups incrementais são usados em conjunto com um backup completo frequente (ex.: um backup completo semanal, com incrementais diários).

- A vantagem principal em usar backups incrementais é que rodam mais rápido que os backups completos. Isto acontece porque um backup incremental só efetivamente copia os arquivos que foram alterados desde o último backup efectuado (incremental ou diferencial).
- Outra vantagem dessa solução é a economia tanto de espaço de armazenamento quanto de tempo de backup, já que o backup só será feito dos arquivos alterados desde o último backup.
- A principal desvantagem dos backups incrementais é que para restaurar um determinado arquivo, pode ser necessário procurar em um ou mais backups incrementais até encontrar o arquivo. Para restaurar um sistema de arquivo completo, é necessário restaurar o último backup completo e todos os backups incrementais subsequentes. Este método gasta muito tempo recriando a estrutura original, que se encontra espalhada entre vários backups diferentes, o que pode tornar o processo lento e suscetível a riscos, se houver algum problema em um dos backups incrementais entre o backup completo e o último backup incremental.

Numa tentativa de diminuir a necessidade de procurar em todos os backups incrementais, foi implementada uma táctica ligeiramente diferente. Esta é conhecida como backup diferencial.

Backups Diferenciais

Da mesma forma que o backup incremental, o backup diferencial também só copia arquivos alterados desde o último backup. No entanto, a diferença deste para o integral é o de que cada backup diferencial mapeia as alterações em relação ao último backup completo. Como o backup diferencial é feito com base nas alterações desde o último backup completo, a cada alteração de arquivos, o tamanho do backup vai aumentando, progressivamente.

Em determinado momento pode ser necessário fazer um novo backup completo pois nesta situação o backup diferencial pode muitas vezes ultrapassar o tamanho do backup integral.

Em relação ao backup completo, ele é mais rápido e salva espaço e é mais simples de restaurar que os backups incrementais.

- A desvantagem é que vários arquivos que foram alterados desde o último backup completo serão repetidamente copiados. Backups diferenciais são similares aos backups incrementais pois ambos podem fazer backup somente de arquivos modificados.

No entanto, os backups diferenciais são acumulativos, em outras palavras, no caso de um backup diferencial, uma vez que um arquivo foi modificado, este continua a ser incluso em todos os backups diferenciais (obviamente, até o próximo backup completo). Isto significa que cada backup diferencial contém todos os arquivos modificados desde o último backup completo, possibilitando executar uma restauração completa somente com o último backup completo e o último backup diferencial.

Assim como a estratégia utilizada nos backups incrementais, os backups diferenciais normalmente seguem a mesma tática: um único backup completo periódico seguido de backups diferenciais mais frequentes. O efeito de usar backups diferenciais desta maneira é que estes tendem a crescer um pouco ao longo do tempo (assumindo que arquivos diferentes foram modificados entre os backups completos). Isto posiciona os backups diferenciais em algum ponto entre os backups incrementais e os completos em termos de velocidade e utilização da mídia de backup, enquanto geralmente oferecem restaurações completas e de arquivos mais rápidas (devido o menor número de backups onde procurar e restaurar). Dadas estas características, os backups diferenciais merecem uma consideração cuidadosa.

Backups Delta

Este tipo de backup armazena a diferença entre as versões correntes e anteriores dos arquivos. Este tipo de backup começa a partir de um backup completo e, a partir daí, a cada novo backup são copiados somente os arquivos que foram alterados enquanto são criados hardlinks

para os arquivos que não foram alterados desde o último backup. Esta é a técnica utilizada pela Time Machine da Apple e por ferramentas como o rsync.

- A grande vantagem desta técnica é que ao fazer uso de hardlinks para os arquivos que não foram alterados, restaurar um backup de uma versão actual é o equivalente a restaurar o último backup, com a vantagem que todas as alterações de arquivos desde o último backup completo são preservadas na forma de histórico.
- A desvantagem deste sistema é a dificuldade de se reproduzir esta técnica em unidades e sistemas de arquivo que não suportem hardlinks.

Alguns autores citam essas formas de backup apresentadas anteriormente como estratégias, uma vez que prescindem de análise para sua utilização. Contudo, tais autores acrescentam, como sendo espécies de backup, os seguintes:

- **Backups Operacionais:** Representam as cópia de segurança realizadas das actividades operacionais e ficam acondicionadas próximas ou dentro do Ambiente de Trabalho, pois podem ser requeridas com certa urgência para continuar as atividades operacionais.
- **Backups Contingenciais:** São cópias guardadas fora do Ambiente de Trabalho a fim de serem utilizadas para resolver Situações de Contingência.
- **Backups Legais:** Constituem-se das cópias de segurança realizadas para fins de manutenção de Registros Históricos, Fiscais ou para Auditoria, uma vez que podem ser solicitadas para esclarecimentos ou por solicitações legais ou fiscais.

Conclusão

Os backups são procedimentos inquestionáveis ao nível de Base de Dados, pois quem trabalha com informação deve ter em conta o conceito de backup (copia de segurança), desde a informação mais elementar possível.

Nesta secção, abordamos os diversos tipos de backup a ter em conta quando estamos a trabalhar com as Base de dados.

Avaliação da actividade

Verifique a sua compreensão!

1. O que entendes por backup?
2. Dê exemplo de uma forma de backup mais elementar possível no seu dia a dia?
3. Em que ocasião podemos implementar o backup incremental?

Actividades de Aprendizagem

Actividade 4.4. Recovery

Contextualização

A recuperação completa das BD é, via de regra, para os utilizadores, uma tarefa simples. Contudo devemos lembrar que, normalmente, dependendo do tamanho da BD, é algo bem demorado devendo ser realizado em horários de janela de processamento, em que o sistema não seja utilizado.

Devemos lembrar que, ao ser baixado integralmente o backup completo, todos os dados anteriores presentes no backup serão substituídos pelos do backup, ao ser feito o recovery. Às vezes, porém, além da BD principal, outras tabelas ou arquivos, que não apresentaram problemas e estão mais actualizadas que as do backup realizado, poderiam representar problemas se fossem substituídas por versões mais antigas.

Assim, a recuperação completa dos dados, operacionalmente, deve requerer uma análise detalhada e um cuidado em seu planeamento e realização, uma vez que, como vimos, ao ser recuperado completamente, a BD irá substituir o anterior, integralmente.

Dessa forma, a fim de evitar as perdas de alguns dados que foram actualizados no período de "janela" (espaço) entre um backup e outro, podem ser utilizadas algumas estratégias para minimizar os problemas.

Para tanto, é necessário que os utilizadores sejam informados de como são realizados os procedimentos e possam tentar verificar se existem dados e arquivos em bom estado, com datas de actualização posteriores às da realização dos últimos backups, os quais podem ser, pelo próprio utilizador, copiados para outras áreas ou mídias de armazenamento e, depois da restauração, sobrepor alguns dos dados restaurados.

Uma das formas de minimizar tais problemas é a utilização de rotinas que contenham a utilização de comandos de comparação de valores e de arquivos, que variam de acordo com o Sistema de Recovery ou SGBD utilizados, em que a restauração de todos os arquivos poderá ser realizada em outra pasta/directório que não os de uso e tais rotinas verificarão, por exemplo, se os arquivos são idênticos, significando que o arquivo não precisa ser substituído e que está bom, ou se a data do arquivo que está no backup é anterior ao do arquivo presente no directório e, se este estiver em boas condições, não será trocado.

A restauração parcial de backups implica, basicamente, as especificidades dos softwares utilizados para realizar os backups. Como a maioria destes apresenta registros de log internos bem detalhados, é possível permitir aos utilizadores responsáveis a escolha específica do que se quer restaurar, fornecendo um estado operacional consistente e utilizável. Contudo, esse processo de escolha pode ser normalmente, um pouco complicado, uma vez que pode depender de intervenção manual ou da confecção de rotinas.

Shadow paging (paginação sombra)

Sombra de paginação é uma solução para durabilidade e atomicidade em base de dados, mas não é tão popular como usando o log de write-ahead.

Aqui está uma analogia para explicar paginação sombra: Suponha que você precisa editar uma página da web em seu site (page.html), para fazer grandes mudanças no design e no conteúdo. Mas você está preocupado que, se as pessoas visitarem a página enquanto você está trabalhando nisso, ele verá HTML quebrado.

A solução é fazer uma cópia da página web com um novo nome (como page2.html), e editar esse novo arquivo. Então você pode estar livre para fazer muitas mudanças e testá-los. Ninguém mais vai ver a página, porque nenhuma outra parte do seu web site tem links para Page2.html.

Quando você tiver completamente feito as suas edições, você renomea os arquivos de modo que o primeiro arquivo é page.old.html, e o arquivo editado se torna page.html.

Nessa analogia, page2.html é como a página de sombra em um DBMS. Actualizações de dados são feitas em uma cópia de uma página. Quando as actualizações são feitas, as referências à página antiga são alteradas para fazer referência à nova página, e isso conta como uma ação atômica.

O propósito de fazer isso é preservar a página original, apenas no caso de as actualizações receberem um erro, e também para fazer a actualização aparecer atômica para outros utilizadores do banco de dados.

Fase de recuperação

Normalmente, existem quatro fases quando se trata de uma recuperação de dados realizada com sucesso, apesar de que pode variar dependendo do tipo de corrupção de dados e da recuperação exigida.

Fase 1: reparo do disco rígido: Reparar o disco rígido de forma que ele funcione de alguma forma, ou pelo menos em um estado adequado para leitura de seus dados. Por exemplo, se as cabeças estiverem com defeito elas devem ser substituídas; se a PCB estiver com falha, então ela precisa ser consertada ou substituída; se o motor do eixo (spindle) estiver danificado os pratos e as cabeças devem ser movidas para uma nova unidade.

Fase 2: gerar imagem da unidade em uma nova unidade ou em um arquivo de imagem de disco: Quando um disco rígido falha, a importância de obter-se os dados extraindo-os do disco é prioridade. Quanto mais tempo uma unidade for utilizada, maior será a probabilidade de ocorrência de perda de dados. Criar uma imagem da unidade garantirá que haverá uma cópia secundária dos dados em outro dispositivo, no qual estará mais seguro para realização de procedimentos de teste e recuperação sem prejudicar a fonte.

Fase 3: recuperação lógica de arquivos, partições, MBR e MFT: Após a unidade ter sido clonada para uma nova unidade, é adequado tentar recuperar os dados perdidos. Se a unidade estiver falhando logicamente, há um número de razões par isto. Usando o clone pode ser possível reparar a tabela de partição, o MBR e o MFT com o objetivo de ler a estrutura de dados do sistema de arquivos e recuperar os dados armazenados.

Fase 4: reparo de arquivos danificados que foram recuperados: Danos em dados podem ser causando quando, por exemplo, um arquivo é escrito em um sector na unidade que tinha sido danificado. Esta é a causa mais comum em uma unidade com falha, significando que os dados precisam ser reconstruídos para se tornarem legíveis. Documentos corrompidos podem ser recuperados por vários métodos de software ou por reconstrução manual do documento usando um editor hexadecimal.

Conclusão

Quem trabalha com backups tem em mente a necessidade de um dia fazer o recovery, pois trata-se de conceitos que vivem associados. Nesta actividade, abordamos o conceito de recovery, as suas fases de recuperação e o termo pouco conhecido ate então (Shadow paging).

Avaliação da actividade

Verifique a sua compreensão!

1. Defina o termo recovery?
2. Entre na Internet e liste quatro softwares que permitam fazer o recovery.
3. Faça um resumo sobre as fases do recovery.

Resumo da Unidade 4

Nesta unidade abordamos o conceito de Segurança de Base de Dados, onde procuramos trazer os tipos de segurança; o controlo de acesso à Base de dados, o Backup e Recovery. No final de cada Unidade, o estudante deve procurar responder as questões de avaliação, por forma a medir o seu nível de compreensão.

Aconselhamos também que faça resumos sobre cada uma das actividades aqui apresentadas e pesquise sobre as mesmas por forma a ter uma visão mais ampla sobre a segurança de base de dados.

.

Avaliação da Unidade 4

Verifique a sua compreensão!

1. Quais são os aspectos a ter em conta na segurança de base de Dados?
2. Crie uma base de dados chamada UNIVERSIDADE e dê todos os privilégios a utilizadora KIARA, Cujo password é ABC.
3. Este tipo de backup armazena a diferença entre as versões correntes e anteriores dos arquivos. Este tipo de backup começa a partir de um backup completo e, a partir daí, a cada novo backup são copiados somente os arquivos que foram alterados enquanto são criados hardlinks para os arquivos que não foram alterados desde o último backup. De que backup se trata?
4. Existe uma diferença entre restore e recovery?

Soluções

1. A segurança em Banco de Dados é constituída pelo estudo do conjunto de medidas, por mecanismos e políticas de segurança que se destinam a prover dispositivos, a fim de que sejam mantidos os aspectos de disponibilidade, de confidencialidade e integridade dos dados, além de fornecer protecção aos sistemas contra possíveis acidentes, falhas ou ataques perpetrados interna ou externamente às empresas.
2. Create database UNIVERSIDADE;

   Create user KIARA identified by ABC;

   GRANT all ON UNIVERSIDADE TO kiara [WITH GRANT OPTION]
3. Leia o texto acima sobre backup.
4. O recovery consiste no processo de recuperação enquanto que o restore consite no processo de devolução.

Leituras e outros Recursos

- Documentos acedidos via Internet no dia 27.02.2016
- Naveen Prakash, Introduction to database management", TMH, 1993.
- Bobrowski, " client server architecture and introduction to oracle 7", 1996
- http://diretorio.shopsul.net/download/Base de_dados/tecnicas_de_backup_e_recovery.pdf
- http://www.diegomacedo.com.br/backup-conceito-e-tipos/
- https://www.quora.com/What-is-shadow-paging-in-dbms

Avaliação do Curso

Avaliação Sumativa

1. Qual é a filosofia do surgimento do modelo OR?
2. Porque utilizar a BDOR?
3. O que é um Objecto complexo não-estruturado?
4. Defina o termo consulta?
5. Defina optimização de consulta.
6. Defina heurística?
7. Defina Base de Dados Distribuída?
8. Quando é que podemos considerar uma réplica síncrona?
9. Em que consiste o data mining?
10. Qual a diferença entre restore e recovery?
11. Quais são os aspectos a ter em conta na segurança de base de Dados?
12. Crie uma base de dados chamada UNIVERSIDADE e dê todos os privilégios a utilizadora KIARA, Cujo password é ABC.
13. Este tipo de backup armazena a diferença entre as versões correntes e anteriores dos arquivos. Este tipo de backup começa a partir de um backup completo e, a partir daí, a cada novo backup são copiados somente os arquivos que foram alterados enquanto são criados hardlinks para os arquivos que não foram alterados desde o último backup. De que backup se trata?

Correção da Avaliação Sumativa

1. Sol 1: O modelo de base de dados surgiu numa tentativa de unir as vantagens de ambas as tecnologias anteriores, as BDOR modelam objectos armazenados em tabelas, ou seja, utilizam as tabelas do modelo relacional, mas nelas são armazenados objectos, com seus atributos e comportamentos, unindo assim os paradigmas
2. Sol 2: Alguns motivos que levam a utilizaçãoda BDOR são: A incapacidade do modelo relacional básico de resolver os desafios e atender as necessidades das novas aplicações; A capacidade de armazenar novos tipos de dados; Fornecem suporte para consultas complexas sobre dados complexos e Atendem aos requisitos das novas aplicações e da nova geração de aplicações de negócios
3. Sol 3: É um objecto que utiliza um tipo que não é padrão na BD e podem ocupar uma grande·área de armazenamento. Estes objectos são do tipo BLOB, que significa Binary Large Object, e podem armazenar, por exemplo, os dados referentes a uma imagem bitmap, um texto formatado, uma musica entre outros.
4. Sol 4. Uma consulta (query) é uma especificação de alto nível de um conjunto de objectos de interesse da base de dados . Geralmente é especificada em uma linguagem especial (Data Manipulation Language - DML) que descreve o que o resultado deve conter .
5. Sol 5. Optimização de consulta (Query optimization) tem o objetivo de escolher uma estratégia de execução eficiente para a execução da consulta
6. Sol 6. A heurística é uma das opções que possibilita a optimização de uma consulta, ela utiliza regras que modificam a representação interna da consulta e melhoram o desempenho esperado para a execução dessa consulta.
7. Sol 7: Uma Base de dados distribuída consiste numa colecção de uma ou mais base de dados logicamente integradas e distribuídas por vários computadores ligados por uma rede física
8. Sol 8: Numa replicação síncrona, cada transacção é dada como concluída quando todos os nós confirmam que a transacção local foi bem sucedida
9. Sol 9: data mining é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.
10. Sol 10: O recovery consiste no processo de recuperação enquanto que o restore consite no processo de devolução.

11. Sol 11: A segurança em Banco de Dados é constituída pelo estudo do conjunto de medidas, por mecanismos e políticas de segurança que se destinam a prover dispositivos, a fim de que sejam mantidos os aspectos de disponibilidade, de confidencialidade e integridade dos dados, além de fornecer protecção aos sistemas contra possíveis acidentes, falhas ou ataques perpetrados interna ou externamente às empresas.

12. Sol 12: Create database UNIVERSIDADE;

    Create user KIARA identified by ABC;

    GRANT all ON UNIVERSIDADE TO kiara [WITH GRANT OPTION]

13. Sol 13: Backups Delta

Sobre o Autor

Sou Aurélio Armando Pires Ribeiro, docente da Universidade Pedagógica, na Escola Superior Técnica, Departamento de Informática, onde lecciono várias disciplinas, particularmente Tecnologias de Base de Dados e Laboratório de Base de Dados. Também trabalho no Centro de Educação Aberta e à Distância, da mesma universidade, sendo o responsável pelo departamento tecnologias de informação e comunicação para a educação à distância.

Fiz os cursos de bacharelato e Licenciatura em Ensino de Informática na Universidade Pedagógica, assim como o mestrado em Informática Educacional, curso administrado pela Universidade Pedagógica em parceria com a Universidade Federal do Rio Grande do Sul do Brasil.

Tenho certa experiência em educação a distância, pois por além de ser tutor nesta modalidade de ensino, desempenho actividades de edição e maquetização de materiais auto-instrucionais, faço pesquisa em tecnologias para apoio à educação a distância assim como dou formação a docentes e estudantes em ambientes virtuais de aprendizagem.

Este manual foi desenvolvido de acordo com o currículo da Africano Universidade Virtual (UVA), numa equipa composta por falantes de Inglês, Francês e Português. Espero que goste do assunto aqui abordado.

Bom estudo!

Aurélio Ribeiro, MEd

Desenvolvedor e editor de conteúdo em Português

Email: lellopires@gmail.com

Referências do Modulo

- Tecnologia de Bases de Dados (3ª Edição), José Luís Pereira, Editora: FCA - Editora Informática, 1998, ISBN: 9789727221431
- Introdução a Sistemas de Base de Dados, DATE, C. J.. Elsevier Editora, 2004.
- HECTOR GARCIA-MOLINA, JEFFREY D. ULLMAN, JENNIFER D. WIDOM: Database Systems: The Complete Book, , Prentice Hall; 1st edition (October 2, 2001), ISBN: 0130319953
- RAGHU RAMAKRISHNAN, JOHANNES GEHRKE, MCGRAW-HILL: Database Management Systems, 3 edition (August 14, 2002), ISBN: 0072465638.
- C.J. Date, " An introduction to database systems", (3rd ed Narosa publishers, 1985), 1997 (reprint)
- Henry F.korth, Abraham, " database system concepts", McGraw hill Inc., 1997.
- Naveen Prakash, Introduction to database management", TMH, 1993.
- Bobrowski, " client server architecture and introduction to oracle 7", 1996
- http://diretorio.shopsul.net/download/Base_de_dados/tecnicas_de_backup_e_recovery.pdf
- http://www.diegomacedo.com.br/backup-conceito-e-tipos
- https://www.quora.com/What-is-shadow-paging-in-dbms
- https://pt.wikipedia.org/wiki/ACID
- http://www.devmedia.com.br/conceitos-transacoes-e-log-transaction/10089
- https://technet.microsoft.com/pt-BR/library/ms190612(v=sql.105).aspx
- http://www.inf.unioeste.br/~clodis/BDII/BDII_Modulo_1.pdf

Sede da Universidade Virtual africana

The African Virtual University
Headquarters

Cape Office Park

Ring Road Kilimani

PO Box 25405-00603

Nairobi, Kenya

Tel: +254 20 25283333

contact@avu.org

oer@avu.org

Escritório Regional da Universidade Virtual Africana em Dakar

Université Virtuelle Africaine

Bureau Régional de l'Afrique de l'Ouest

Sicap Liberté VI Extension

Villa No.8 VDN

B.P. 50609 Dakar, Sénégal

Tel: +221 338670324

bureauregional@avu.org