

Resumão aulas : 1 (RAID), 2 (índices), 4(processamento de consulta), 8 (transações), 9 (recuperação de dados) e 12 (concorrência).

Resumo da Aula 1 – Armazenamento, RAID e Arquivos:

Armazenamento físico: classificado de acordo com custo, velocidade e confiabilidade, incluindo cache, memória principal, memória flash, disco magnético, disco óptico e fitas magnéticas.

Discos magnéticos: consistem em placas circulares planas com dados gravados em sua superfície, divididas em trilhas e setores. Possuem cabeças de leitura-escrita, controladoras de disco e interfaces como ATA, SATA, SCSI e SAS.

Medidas de desempenho dos discos: capacidade, tempo de acesso, tempo de busca médio, tempo de latência rotacional médio, tempo de transferência, taxa de transferência e confiabilidade.

RAID (Redundant Arrays of Independent Disks): melhora a confiabilidade por redundância e o desempenho por paralelismo. Os níveis de RAID incluem espelhamento, espalhamento e combinações de ambos.

Escolha do nível de RAID: depende de fatores como custo, desempenho, segurança dos dados, recuperação de falhas e capacidade de armazenamento.

Recomendações: RAID 0 para recuperação rápida, RAID 1 para melhor desempenho, RAID 5 para baixa taxa de acesso com grande quantidade de dados.

O que é RAID?

RAID (Redundant Array of Independent Disks) é uma tecnologia de armazenamento que combina múltiplos discos rígidos para melhorar o desempenho, a confiabilidade ou a capacidade de armazenamento de um sistema. O objetivo principal do RAID é fornecer maior tolerância a falhas e maior rendimento em comparação com um único disco rígido.

Existem vários níveis de RAID, cada um com seu próprio método de organização e redundância dos dados. Alguns dos níveis de RAID mais comuns são:

RAID 0: Também conhecido como "striping", divide os dados em blocos e grava esses blocos em vários discos simultaneamente. Isso melhora o desempenho de leitura e gravação, mas não fornece redundância ou tolerância a falhas.

RAID 1: Conhecido como "espelhamento", consiste em duplicar os dados em dois discos separados. Isso fornece redundância, pois os dados são idênticos em ambos os discos. Se um disco falhar, os dados ainda estarão disponíveis no outro disco.

RAID 5: Distribui os dados e a paridade (informações de verificação de redundância) em vários discos. A paridade é usada para reconstruir dados em caso de falha de disco. O RAID 5 requer pelo menos três discos para ser implementado e oferece uma combinação de desempenho e tolerância a falhas.

RAID 10: Também conhecido como RAID 1+0, combina as características de espelhamento (RAID 1) e striping (RAID 0). Os dados são espelhados em discos separados e, em seguida, os conjuntos espelhados são combinados em um conjunto maior usando striping. Isso fornece tanto redundância quanto desempenho.

Além desses, existem outros níveis de RAID, como RAID 2, RAID 3, RAID 4, RAID 6, RAID 50, entre outros, cada um com suas próprias características e usos específicos. A escolha do nível de RAID adequado depende dos requisitos de desempenho, confiabilidade e capacidade de armazenamento do sistema.

Resumo da Aula 2- ÍNDICES

Nessa aula sobre índices em bancos de dados, a professora abordou os seguintes pontos:

1. Conceitos preliminares:
 - Consulta como recuperação de uma pequena fração de registros.
 - Acesso eficiente aos dados em vez de ler um por um.
 - Necessidade de criar estruturas especiais para associar os registros.
 - Introdução aos índices como solução.
2. Índices ordenados:
 - Estrutura de índice que associa uma chave de busca a um registro.
 - Índices primários ou de agrupamento, quando os registros e a chave de busca estão ordenados sequencialmente.
 - Índices densos, que possuem um registro de índice para cada valor de chave de busca.
 - Índices esparsos, que possuem registros de índice apenas para alguns valores de chave de busca.
3. Inserção e exclusão em índices densos e esparsos:
 - Processo de inserção e atualização dos registros de índice quando novos registros são adicionados.
 - Processo de exclusão e atualização dos registros de índice quando registros são removidos.
4. Índices secundários:
 - Índices adicionais que têm como chave de busca um atributo diferente da chave primária.
 - São densos e possuem um ponteiro para cada registro.
 - Os blocos de registros são ordenados de acordo com a chave de busca do índice agrupado.
5. Índices esparsos multiníveis:

- Solução para lidar com índices esparsos que não cabem completamente na memória principal.
 - Utiliza múltiplos níveis de indexação para melhorar o desempenho.
6. Problemas dos índices ordenados:
- Degradação de desempenho à medida que a estrutura de índice cresce.
 - Necessidade de explorar outras estruturas de organização não ordenadas.

Esses são os principais conceitos e tópicos abordados nessa aula sobre índices em bancos de dados.

Resumo da Aula 4– PROCESSAMENTO DE CONSULTAS

A aula 4 aborda o processamento de consultas em bancos de dados. O objetivo é explicar como um Sistema de Gerenciamento de Banco de Dados (SGBD) processa consultas e quais fatores estão envolvidos nesse processo.

O processamento de consultas tem como finalidade apoiar a extração de dados de um banco de dados. Para isso, envolve diversas atividades, como a tradução de expressões de alto nível para implementação em nível físico, otimizações e avaliação das consultas.

Durante o processamento de consultas, as consultas SQL são traduzidas em expressões de Álgebra Relacional. Para cada expressão, existem várias formas de representação algébrica e vários algoritmos possíveis. Por exemplo, para realizar a seleção, é possível usar índices ou algoritmos de busca sequencial.

Para avaliar uma consulta, é necessário anotar cada operação e como ela será avaliada, considerando o uso de índices e algoritmos. Essas anotações resultam em uma avaliação primitiva (AP) que pode ser agrupada em um pipeline, permitindo execuções em paralelo. A sequência de APs é chamada de Plano de Execução de Consulta (PEC).

Um SGBD trabalha para minimizar os custos envolvidos no processamento de consultas. O custo mais crítico geralmente é o número de acessos a discos, pois o acesso a disco é uma operação lenta em comparação com o acesso à memória.

Durante a otimização de consultas, o objetivo é selecionar o plano de avaliação de consulta mais eficiente. Isso envolve encontrar expressões equivalentes em Álgebra Relacional que sejam mais eficientes e escolher a estratégia adequada para processar a consulta, incluindo algoritmos e índices.

Para estimar o custo de um Plano de Execução de Consulta (PEC), é necessário consultar as estatísticas do banco de dados. As estatísticas incluem informações como o número de tuplas em uma relação, o número de blocos que contêm as tuplas, o tamanho das tuplas, o número médio de registros que satisfazem uma condição de igualdade em um atributo, entre outros. Além disso, é necessário conhecer os possíveis algoritmos utilizados para cada operação.

Alguns algoritmos comuns para a operação de seleção incluem busca linear, busca binária e utilização de índices primários ou secundários. Cada algoritmo tem seu próprio custo associado, que pode ser estimado com base nas estatísticas do banco de dados.

É importante ressaltar que as explicações acima foram baseadas nos trechos fornecidos, e pode haver mais informações relevantes na aula completa.

Recomendo que você revise todo o conteúdo fornecido e consulte materiais adicionais ou exemplos práticos para se preparar para a prova.

Aula 8 – TRANSAÇÕES

A aula 8 abordou o tema de transações em bancos de dados. Transação é uma unidade de execução do programa que acessa e atualiza vários itens de dados. Geralmente, é iniciada por um programa do usuário, escrito em uma linguagem de alto nível, e é delimitada pelas instruções "begin transaction" e "end transaction".

Para garantir a integridade dos dados, um sistema de gerenciamento de banco de dados (SGBD), como o PostgreSQL, deve manter as propriedades ACID: Atomicidade, Consistência, Isolamento e Durabilidade.

As propriedades ACID são definidas da seguinte forma:

- Atomicidade: Todas as operações de uma transação devem ser executadas ou nenhuma delas. Isso garante que as alterações no banco de dados sejam tratadas como uma unidade indivisível.
- Consistência: A execução de uma transação isolada deve manter a consistência dos dados. Por exemplo, no caso de uma transferência de dinheiro entre duas contas bancárias, a soma dos saldos das contas deve permanecer inalterada.
- Isolamento: Uma transação não deve "perceber" a existência de outras transações concorrentes. Isso significa que, dada uma determinada transação, ou ela terminou antes das outras transações começarem ou começou depois que as outras transações terminaram.
- Durabilidade: Após o término de uma transação, as mudanças feitas no banco de dados devem ser permanentes, mesmo em caso de falhas no sistema.

Um exemplo clássico de transação é a transferência de dinheiro entre duas contas bancárias. É importante garantir que a consistência, atomicidade, durabilidade e isolamento sejam mantidos durante essa operação.

Além disso, foram apresentados os diferentes estados de uma transação: ativa, parcialmente confirmada, falha, abortada e confirmada. Em caso de falha, a transação pode ser revertida e o banco de dados é restaurado ao estado anterior ao início da transação.

A aula 8 menciona dois tópicos relevantes para correlação: "Transação – Serial por Conflito" e "Grafo de Precedência". Vou explicar brevemente cada um deles e como estão relacionados.

1. Transação – Serial por Conflito:

- Neste contexto, uma transação refere-se a uma sequência de operações que são executadas em um banco de dados.
- A abordagem "Serial por Conflito" é um método para garantir a corretude de transações concorrentes.
- A ideia básica é que as transações sejam executadas de forma serializada, uma após a outra, a fim de evitar conflitos e inconsistências de dados.
- Esse método é baseado no bloqueio (locking) de itens de dados, para evitar que múltiplas transações acessem e modifiquem os mesmos dados simultaneamente, o que poderia resultar em resultados indesejados.
- O Serial por Conflito é um dos métodos de controle de concorrência utilizados em bancos de dados, visando garantir a consistência e isolamento das transações.

2. Grafo de Precedência:

- O Grafo de Precedência é uma estrutura gráfica usada para representar as dependências entre as operações de uma transação.
- Em um grafo de precedência, cada operação é representada por um nó, e as dependências entre as operações são representadas por arestas direcionadas.
- Uma aresta do nó A para o nó B indica que a operação A deve ser executada antes da operação B.
- O grafo de precedência pode ser usado para analisar a corretude de uma serialização de transações.
- Se o grafo de precedência não contiver ciclos, é possível encontrar uma serialização equivalente às transações concorrentes.
- No entanto, se houver um ciclo no grafo de precedência, significa que não existe uma serialização equivalente e pode ocorrer um impasse (deadlock) nas transações.

Portanto, a correlação entre esses dois tópicos é que o método de "Transação – Serial por Conflito" utiliza o conceito de "Grafo de Precedência" para analisar a ordem de execução das transações e evitar conflitos. O grafo de precedência

permite identificar as dependências entre as operações e determinar se uma serialização é possível ou se pode ocorrer um impasse.

Aula 9 – Recuperação dos dados

A aula aborda o tema "Recuperação dos dados" e apresenta diferentes conceitos e técnicas relacionados a esse processo. Vou fornecer uma explicação resumida de cada tópico mencionado na aula:

1. Diagrama de Estados:
 - O diagrama de estados é uma representação gráfica que mostra os diferentes estados pelos quais uma transação pode passar durante sua execução.
 - Os estados mencionados na aula são: Ativa, Parcialmente Confirmada, Falha Abortada e Confirmada.
 - Esses estados refletem o progresso de uma transação desde o início até sua conclusão ou falha.
2. Implementação de Atomicidade e Durabilidade:
 - Nesse tópico, é discutida a responsabilidade do componente de gerenciamento de recuperação em um sistema de gerenciamento de banco de dados (SGBD).
 - É mencionado o esquema de cópia de sombra, onde uma transação cria uma cópia completa do banco de dados antes de fazer as atualizações.
 - Em caso de falha ou aborto da transação, a cópia é excluída, enquanto, se a transação for confirmada, as alterações são confirmadas no disco.
 - A implementação de atomicidade e durabilidade é essencial para garantir a consistência e a durabilidade das transações.
3. Facilidade de Recuperação:
 - Nessa seção, discute-se a importância de garantir a recuperação adequada das transações em caso de falhas.
 - São mencionados os tipos de schedules aceitáveis para garantir a recuperação de falhas, como os schedules recuperáveis versus não recuperáveis e em cascata versus não em cascata.

- Um schedule recuperável é aquele em que nenhuma transação é confirmada antes que todas as transações que gravaram os dados lidos por ela tenham sido confirmadas.
- Evitar os rollbacks em cascata é preferível para evitar desperdício de trabalho e manter a consistência do banco de dados.

4. Schedules Estritos:

- Um schedule estrito é um tipo de schedule que garante a atomicidade e evita problemas de atualização tardia.
- Em um schedule estrito, uma transação só pode ler ou atualizar um dado depois que todas as transações que o atualizaram foram concluídas.
- Isso evita que uma transação leia um dado desatualizado ou inconsistente.

5. Recuperação:

- Nesta parte, é discutido o processo de recuperação de falhas no contexto de um banco de dados.
- São mencionados os dois métodos principais: modificação adiada e modificação imediata.
- A recuperação envolve a utilização de logs, que registram as atualizações feitas no banco de dados durante as transações.
- Através dos registros no log, é possível desfazer ou refazer as operações necessárias para restaurar o banco de dados a um estado consistente, atômico e durável.

6. Checkpoints:

- Os checkpoints são pontos de verificação periódicos que reduzem a sobrecarga do sistema durante o processo de recuperação.
- Eles permitem determinar a partir de que ponto do log é necessário iniciar a recuperação após uma falha.
- Através dos checkpoints, apenas as transações que

Aula 12 – concorrência

A aula em questão aborda o tema de concorrência em bancos de dados, mais especificamente as técnicas de controle de concorrência. No mundo real, é comum que as instruções sejam executadas em paralelo, e o objetivo do sistema de gerenciamento de banco de dados (SGBD) é garantir a correta execução das transações concorrentes.

Uma das técnicas utilizadas para controlar a concorrência é o bloqueio dos itens de dados. Essa técnica impede que múltiplas transações acessem os mesmos itens concorrentemente. Existem dois tipos de bloqueio: binário e compartilhado/exclusivo.

No bloqueio binário, cada item de dado possui um estado de bloqueio: 1 para bloqueado e 0 para desbloqueado. Uma transação precisa esperar até que um item bloqueado seja liberado antes de acessá-lo.

No bloqueio compartilhado/exclusivo, várias transações podem acessar o mesmo item se todas forem apenas para leitura (bloqueio compartilhado). Porém, se uma transação for alterar o item, ela deve ter o bloqueio exclusivo, permitindo apenas o seu acesso.

Para aumentar a concorrência, é possível usar bloqueio compartilhado para leituras e bloqueio exclusivo para escritas. Dessa forma, várias transações podem ler um item simultaneamente, desde que nenhuma esteja escrevendo nele.

É importante ressaltar que os bloqueios não garantem a serialização das transações nem previnem deadlocks (situação em que duas ou mais transações ficam esperando indefinidamente uma pela outra). Por isso, é necessário utilizar um protocolo adicional, como o Two Phase Commit (2PC), que consiste em duas fases: fase de crescimento e fase de encolhimento.

O protocolo de bloqueio em duas fases com nível de granularidade múltiplo utiliza bloqueios de intenção, indicando quais tipos de bloqueio serão solicitados em cada nó da estrutura de dados. Existem regras que devem ser

seguidas para garantir a compatibilidade dos bloqueios e evitar problemas como bloqueios em conflito e bloqueios em cascata.

No final da aula, são propostos exercícios para aplicar os conceitos apresentados, incluindo a adição de bloqueios em schedules e a identificação de deadlocks.