

TRABALHO PARCIAL

RODRIGO MENEZES

UNIDADE 1

SOBRE A EVOLUÇÃO DE BANCO DE DADOS

Os bancos de dados tiveram origem durante as décadas de 1960 e 1970, quando os computadores digitais começaram a ser desenvolvidos. Edgar Codd, que propôs o modelo de banco de dados relacional em 1960, que se tornou a base para muitos sistemas de gerenciamento de banco de dados que surgiram posteriormente. Com o tempo, o uso de bancos de dados se expandiu em diversas áreas, tornando-se uma parte fundamental da tecnologia da informação. A seguir breve cronologia de sua evolução:

- 1960: Edgar F. Codd propõe o modelo de banco de dados relacional.
- 1966: A IBM lança o Information Management System (IMS), um dos primeiros sistemas de gerenciamento de banco de dados.
- 1970: A IBM desenvolve o System R, um sistema experimental de gerenciamento de banco de dados relacional.
- 1976: A Oracle Corporation é fundada por Larry Ellison, Bob Miner e Ed Oates. A empresa se torna líder no mercado de sistemas de gerenciamento de banco de dados.
- 1980: O modelo de banco de dados relacional se torna amplamente utilizado e substitui outros modelos anteriores, como o modelo hierárquico e o modelo em rede.
- 1990: A Microsoft lança o SQL Server, seu sistema de gerenciamento de banco de dados relacional.
- 2000: A Google começa a usar seu sistema de gerenciamento de banco de dados não relacional, chamado Bigtable.
- 2010: O Apache Cassandra, outro sistema de gerenciamento de banco de dados não relacional, é lançado como um projeto de software livre.
- 2020: Os bancos de dados em nuvem e os bancos de dados distribuídos se tornam cada vez mais populares, permitindo que as empresas armazenem e gerenciem grandes quantidades de dados de forma escalável e eficiente.

SOBRE AS FUNÇÕES DO SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS - SGBD'S

Sistemas de Gerenciamento de Banco de Dados (SGBDs) são ferramentas que permitem que as pessoas organizem e acessem grandes quantidades de informações de maneira segura e eficiente. Eles garantem que muitas pessoas possam acessar as mesmas informações ao mesmo tempo, evitando erros ou problemas. Os SGBDs também ajudam as pessoas a fazerem perguntas complicadas sobre as informações, modificá-las e garantir que elas estejam corretas e seguras. Simplificando, eles são uma maneira útil de organizar e acessar grandes quantidades de informações de maneira fácil e segura.

Sobre os SGBDs e suas características:

- ❖ Acesso seguro aos dados: SGBDs garantem que somente usuários autorizados tenham acesso às informações armazenadas, protegendo-as contra ameaças externas.
- ❖ Backup e recuperação de dados: Os SGBDs permitem a realização de cópias de segurança dos dados, além de possuírem recursos para restaurá-los em caso de perda ou corrupção.
- ❖ Concorrência: SGBDs permitem que múltiplos usuários acessem e modifiquem os dados simultaneamente, evitando conflitos.
- ❖ Consultas complexas: Com os SGBDs, é possível realizar consultas avançadas aos dados, utilizando operações como seleção, projeção e junção.
- ❖ Escalabilidade: Os SGBDs são capazes de gerenciar grandes volumes de dados de forma eficiente, garantindo um bom desempenho.
- ❖ Gerenciamento de transações: SGBDs gerenciam transações de usuários, garantindo que elas sejam concluídas com sucesso ou revertidas em caso de erro ou falha do sistema.
- ❖ Integridade dos dados: Os SGBDs mantêm a integridade dos dados, garantindo que eles estejam sempre corretos e consistentes, utilizando restrições de integridade e validação.

- ❖ Organização e armazenamento de dados: Com os SGBDs, é possível organizar e armazenar os dados em tabelas, o que torna o acesso e a modificação das informações mais fáceis e eficientes.

SOBRE OS SGBDS E SUA ESTRUTURA:

1. Linguagem de Definição de Dados (LDD): é usada para definir a estrutura do banco de dados, criando as tabelas, campos e relacionamentos entre eles.
2. Linguagem de Manipulação de Dados (LMD): é usada para inserir, atualizar e excluir dados no banco de dados, além de permitir a recuperação de informações por meio de consultas.
3. Motor de Banco de Dados: é a parte central do SGBD responsável pelo armazenamento e recuperação dos dados. Ele gerencia as solicitações de acesso aos dados, garantindo a segurança, a integridade e a concorrência de acesso.

O SGBD também possui outros componentes importantes, como o otimizador de consultas, que determina o melhor caminho para executar uma consulta, e o gerenciador de transações, que garante que as transações sejam executadas com sucesso ou revertidas em caso de erro.

ABAIXO ESTÃO ALGUNS EXEMPLOS DE EMPRESAS QUE UTILIZAM DIFERENTES TIPOS DE SGBDS E COMO ESSES SISTEMAS SÃO APLICADOS EM DIFERENTES SETORES:

Oracle: A Oracle é uma das maiores empresas de tecnologia do mundo e oferece uma ampla gama de SGBDs, incluindo o Oracle Database, o MySQL e o MariaDB. A Oracle é amplamente utilizada em setores como finanças, serviços bancários, telecomunicações e saúde, onde a segurança e a escalabilidade são extremamente importantes. Por exemplo, o banco Santander utiliza o Oracle Database para gerenciar suas operações bancárias, enquanto o gigante farmacêutico Pfizer utiliza o Oracle Database para gerenciar sua cadeia de suprimentos.

SQL Server da Microsoft: O SQL Server é um SGBD da Microsoft e é amplamente utilizado em empresas que utilizam a plataforma Windows. O SQL Server é comumente usado em setores como varejo, manufatura e serviços financeiros. Por exemplo, a loja de departamentos Macy's utiliza o SQL Server para gerenciar sua cadeia de suprimentos, enquanto a Delta Airlines utiliza o SQL Server para gerenciar seu sistema de reservas.

Cassandra: O Cassandra é um SGBD distribuído altamente escalável que é amplamente utilizado em empresas de tecnologia que precisam lidar com grandes volumes de dados, como a Netflix, a Apple e a Uber. A Netflix, por exemplo, utiliza o Cassandra para gerenciar sua plataforma de streaming de vídeo, enquanto a Apple utiliza o Cassandra para gerenciar suas contas de usuários.

Bigtable do Google: O Bigtable é um SGBD NoSQL distribuído desenvolvido pelo Google e é usado em empresas que precisam lidar com grandes volumes de dados, como a Google e a Snapchat. A Google utiliza o Bigtable para gerenciar seus serviços de busca e armazenamento em nuvem, enquanto a Snapchat utiliza o Bigtable para gerenciar seus dados de usuários.

SOBRE OS RECURSOS DE BACKUP E RECUPERAÇÃO DE DADOS:

Os Sistemas Gerenciadores de Banco de Dados (SGBDs) são projetados para garantir a integridade e a disponibilidade dos dados armazenados, bem como para proteger esses dados contra perdas acidentais ou maliciosas. Uma das principais funções de um SGBD é gerenciar backups e recuperação de dados.

Backup de dados é o processo de fazer cópias de dados armazenados em um banco de dados em um determinado momento, a fim de proteger esses dados contra perda. Existem vários tipos de backup, incluindo:

1. Backup completo: faz uma cópia completa de todos os dados do banco de dados.
2. Backup incremental: faz uma cópia apenas das alterações que ocorreram desde o último backup, reduzindo o tempo de backup e a quantidade de dados a serem armazenados.
3. Backup diferencial: faz uma cópia das alterações desde o último backup completo, reduzindo a quantidade de dados a serem armazenados em comparação com o backup completo.

A estratégia de backup depende do volume de dados e do nível de segurança exigido pelos dados armazenados. Para bancos de dados críticos, pode ser necessário executar backups frequentes, tanto incrementais quanto diferenciais, a fim de minimizar a perda de dados em caso de falha.

A recuperação de dados é o processo de restaurar dados perdidos ou corrompidos de um backup. Existem vários métodos de recuperação de dados, incluindo:

1. Rollback: desfaz todas as alterações que ocorreram desde o último ponto de verificação válido.
2. Redo: aplica todas as alterações que ocorreram desde o último ponto de verificação válido até o momento atual.
3. Undo: desfaz uma transação específica ou um conjunto de transações.

As estratégias de recuperação também dependem do nível de segurança exigido pelos dados armazenados. A recuperação pode ser automática, com o SGBD detectando e corrigindo automaticamente problemas menores, ou manual, exigindo a intervenção do administrador do banco de dados.

Em geral, os SGBDs oferecem recursos avançados de backup e recuperação de dados para garantir a disponibilidade e a integridade dos dados armazenados. É importante que as empresas que utilizam SGBDs implementem práticas adequadas de backup e recuperação de dados para proteger seus ativos mais valiosos.

SOBRE O TEOREMA CAP

O teorema de CAP, também conhecido como teorema de Brewer, é uma teoria da ciência da computação que afirma que é impossível para um sistema distribuído de computação em nuvem fornecer simultaneamente as três garantias de consistência (C), disponibilidade (A) e tolerância a particionamento de rede (P).

Consistência significa que todos os usuários veem a mesma versão dos dados ao mesmo tempo, independentemente de qual nó do sistema eles estejam acessando. Disponibilidade significa que todos os usuários podem acessar o sistema e obter uma resposta, mesmo que alguns dos nós estejam temporariamente indisponíveis. Tolerância a particionamento de rede significa que o sistema continua a funcionar mesmo se alguns dos nós não estiverem se comunicando.

O teorema afirma que, em um sistema distribuído, somente duas dessas garantias podem ser fornecidas simultaneamente. Isso ocorre porque, em caso de falhas de rede, é necessário escolher entre manter a consistência dos dados ou manter a disponibilidade do sistema. Se o sistema escolher manter a consistência, ele precisará bloquear temporariamente o acesso dos usuários até que todos os nós estejam sincronizados, o que pode afetar a disponibilidade. Se o sistema escolher manter a disponibilidade, ele pode permitir que alguns usuários acessem informações desatualizadas, afetando a consistência.

O teorema de CAP é importante para projetar sistemas distribuídos que atendam às necessidades de um determinado cenário de uso, considerando o equilíbrio entre consistência, disponibilidade e tolerância a particionamento de rede. É comum que os sistemas distribuídos priorizem a disponibilidade e a tolerância a particionamento de

rede em detrimento da consistência, para garantir que o sistema permaneça disponível mesmo em caso de falhas de rede.

O teorema de CAP é uma teoria fundamental em bancos de dados distribuídos, incluindo o MongoDB. Na prática, os engenheiros do MongoDB escolheram otimizar a disponibilidade e a tolerância a particionamento de rede em detrimento da consistência, como muitos outros bancos de dados NoSQL.

O MongoDB é um banco de dados não-relacional que permite alta disponibilidade e tolerância a falhas através de uma arquitetura distribuída. Ele usa um modelo de consistência eventual para garantir a disponibilidade e a tolerância a particionamento de rede, o que significa que, em situações de falhas de rede, as atualizações podem ser propagadas para diferentes nós em momentos diferentes, resultando em dados momentaneamente inconsistentes entre os nós. No entanto, o MongoDB fornece mecanismos para garantir que essas inconsistências sejam eventualmente corrigidas, por exemplo, por meio da replicação de dados entre os nós.

Os desenvolvedores do MongoDB recomendam que as aplicações sejam projetadas com a compreensão de que os dados podem ser momentaneamente inconsistentes entre os nós, e que a aplicação deve ser capaz de lidar com essa situação. Portanto, em sistemas distribuídos que utilizam o MongoDB, é importante garantir que a arquitetura da aplicação leve em consideração a priorização da disponibilidade e tolerância a particionamento de rede em detrimento da consistência.

Vencimento	12 mar por 23:59	Pontos	100	Enviando	um upload de arquivo
Tentativas	0	Tentativas permitidas	1	Disponível	13 fev em 19:00 - 12 mar em 23:59

Critérios que serão considerados para esta entrega:

Unidade 1:

- Fazer uma síntese sobre a evolução dos bancos de dados, citando qual sua importância atualmente para o mundo corporativo.
- Discorrer sobre o papel dos SGBDs, suas principais características e sua estrutura. Comentar sobre o Teorema CAP e como ele é aplicado em banco de dados, principalmente no MongoDB .
- Fazer o download do arquivo a partir do link a seguir: <https://www.kaggle.com/eswarchandt/amazon-music-reviews> ➞

Unidade 2:

- Criar um novo database e uma nova coleção para receber o dataset.
- Importar o novo data set, populando a coleção criada.