

BANCOS DE DADOS NOSQL

Vinícius Kroth e Eduardo Henrique Pereira de Arruda

“

Nenhum sistema é construído **da noite**
para o dia.

”

Daniel Callegari

Conheça o livro da disciplina

CONHEÇA SEUS PROFESSORES 3

Conheça os professores da disciplina.

EMENTA DA DISCIPLINA 4

Veja a descrição da ementa da disciplina.

BIBLIOGRAFIA DA DISCIPLINA 5

Veja as referências principais de leitura da disciplina.

O QUE COMPÕE O MAPA DA AULA? 6

Confira como funciona o mapa da aula.

MAPA DA AULA 7

Links de artigos científicos, informativos e vídeos sugeridos.

RESUMO DA DISCIPLINA 27

Relembre os principais conceitos da disciplina.

AVALIAÇÃO 28

Veja as informações sobre o teste da disciplina.

Conheça seus professores



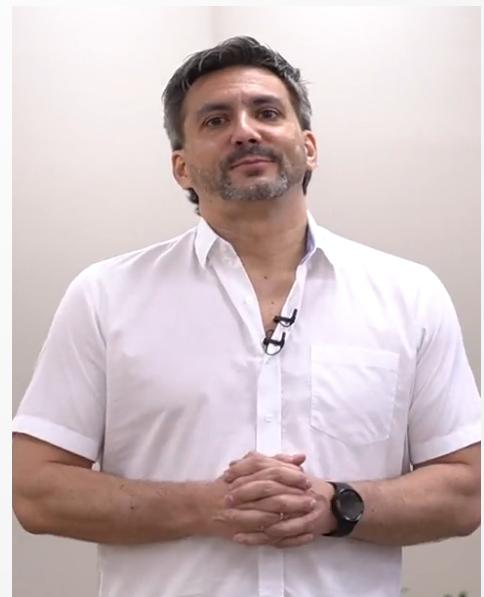
VINÍCIUS KROTH
Professor Convidado

Desenvolvedor de aplicações SOA nas áreas de contabilidade, financeira e de comércio exterior por mais de cinco anos. Referência em assuntos como projeto e desenvolvimento de SOA e Microservices, Java (EE, Spring framework MVC/WebFlux, jUnit e Gradle), SQL e NoSQL Db's (PostgresSQL, MySQL, MongoDB, Redis e Elasticsearch), AWS Cloud computing, Stress/Chaos testing (Gatlin, Jmeter), ferramentas de Desenvolvimento (Jenkins, Docker e Terraform), Telemetria e Observabilidade (Kibana, Datadog).

EDUARDO HENRIQUE PEREIRA DE ARRUDA

Professor PUCRS

Fundador e CEO da Doc.Space Documentos Digitais. Professor da Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS), onde atua desde 1994 em cursos de graduação, pós-graduação e extensão nas áreas de Ciência da Computação, Engenharia de Software e Sistemas de Informação. Possui graduação (1992) e mestrado (1995) em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS) e formações complementares em gestão de TI e Segurança da Informação. Cursa Doutorado no Programa de Pós-Graduação em Ciência da Computação (PPGCC) da PUC-RS. Dedica parte de seu tempo a atividades de incentivo ao empreendedorismo inovador e investe em empresas que adotem modelos de negócio inovadores e escaláveis. Apoia projetos de empreendedorismo social, tendo sido coordenador do projeto Adoções, parceria entre o Poder Judiciário e o Ministério Público do RS com a PUC-RS que, por meio de aplicativo, realiza a aproximação entre candidatos a adoção e crianças e adolescentes em processo de adoção tardia.



Ementa da Disciplina

Introdução aos conceitos e características de Big Data como: volume, velocidade, variedade, validade, volatilidade e valência. Introdução aos conceitos de cluster, domínios, agregados, distribuição, tolerância a falhas e sharding. Estudo do Teorema CAP: consistência (Consistency), disponibilidade (Availability), tolerância de partição (Partition). Introdução a Bancos de dados sem esquema prévio, a Banco de dados baseado em documentos, a Banco de dados chave-valor, a Banco de dados colunar e a Banco de dados baseado em grafos.

Bibliografia da Disciplina

As publicações destacadas têm acesso gratuito.

Bibliografia básica

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de bancos de dados. 7a. Ed. Rio de Janeiro: LTC, 2020.

SILVA, Luiz Fernando; et al. Bando de Dados Não Relacional. Porto Alegre: Sagah, 2021.

PIVERT, Olivier. NoSQL Data Models: Trends and Challenges. Volume 1. Indianapolis: John Wiley & Sons, 2018.

Bibliografia complementar

BALUSAMY, Balamurugan; ABIRAMI, R.; KADRY, S.; Gandomi, A. Big Data: Concepts, Technology, and Architecture. Indianapolis: John Wiley & Sons, 2021.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 7a. Ed. São Paulo: Pearson Education do Brasil, 2018.

YU, Shui ; GUO, Song . Big Data Concepts, Theories, and Applications. Springer, 2016.

SIMON, PI. The visual organization: data visualization, big data, and the quest for better decisions. Hoboken: Wiley, 2014.

SADALAGE, P. J.; FOWLER, M. NoSQL Essencial – Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec, 2013.

O que compõe o Mapa da Aula?

MAPA DA AULA

São os capítulos da aula, demarcam momentos importantes da disciplina, servindo como o norte para o seu aprendizado.



EXERCÍCIOS DE FIXAÇÃO

Questões objetivas que buscam reforçar pontos centrais da disciplina, aproximando você do conteúdo de forma prática e exercitando a reflexão sobre os temas discutidos. Na versão online, você pode clicar nas alternativas.



PALAVRAS-CHAVE

Conceituação de termos técnicos, expressões, siglas e palavras específicas do campo da disciplina citados durante a videoaula.



VÍDEOS

Assista novamente aos conteúdos expostos pelos professores em vídeo. Aqui você também poderá encontrar vídeos mencionados em sala de aula.



PERSONALIDADES

Apresentação de figuras públicas e profissionais de referência mencionados pelo(a) professor(a).



LEITURAS INDICADAS

A jornada de aprendizagem não termina ao fim de uma disciplina. Ela segue até onde a sua curiosidade alcança. Aqui você encontra uma lista de indicações de leitura. São artigos e livros sobre temas abordados em aula.



FUNDAMENTOS

Conteúdos essenciais sem os quais você pode ter dificuldade em compreender a matéria. Especialmente importante para alunos de outras áreas, ou que precisam relembrar assuntos e conceitos. Se você estiver por dentro dos conceitos básicos dessa disciplina, pode tranquilamente pular os fundamentos.

CURIOSIDADES

Fatos e informações que dizem respeito a conteúdos da disciplina.

DESTAQUES

Frases dos professores que resumem sua visão sobre um assunto ou situação.

ENTRETENIMENTO

Inserções de conteúdos para tornar a sua experiência mais agradável e significar o conhecimento da aula.

CASE

Neste item, você relembra o case analisado em aula pelo professor.

MOMENTO DINÂMICA

Aqui você encontra a descrição detalhada da dinâmica realizada pelo professor.

Mapa da Aula

Os tempos marcam os principais momentos das videoaulas.

AULA 1 • PARTE 1

PALAVRAS-CHAVE

JSON: é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida entre sistemas, especificado por Douglas Crockford em 2000, que utiliza texto legível a humanos, no formato atributo-valor.

“ A finalidade principal de um banco de dados é garantir a persistência. ”

PALAVRAS-CHAVE

SGBDS: um Sistema de Gerenciamento de Banco de Dados é um software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados.

01:26



Introdução

O professor Henrique Pereira de Arruda apresenta sua experiência na área, suas formações, qualificações e cargos em seus respectivos trabalhos, explicando aos alunos como as aulas irão se redirecionar, sendo três partes: revisão de conceitos, modelagem relacional e implementação.

07:57



Conceitos básicos

10:37



Um banco de dados deve garantir a durabilidade, a persistência e a segurança. Existem três grandes grupos de usuários: aqueles que podem fazer consultas, aqueles que podem fazer consultas e atualizar os dados e aqueles que podem alterar a estrutura do banco de dados. O banco de dados tem que ser consumido de diversas maneiras em acessos concorrentes, de diversos usuários, com diferentes perfis, diferentes interesses, podendo gerar inúmeros problemas de concorrência.

11:13



18:59



21:48



FUNDAMENTO

Banco de dados relacional

Banco de dados relacional é um banco de dados cujos armazenamentos principais são paralelos. Em 1985, Edgar Frank Codd, criador do modelo relacional, publicou um artigo onde definia 13 regras para que um Sistema Gerenciador de Banco de Dados (SGBD) fosse considerado relacional: regra fundamental, regra da informação, regra da garantia de acesso, tratamento sistemático de valores nulos, catálogo dinâmico online baseado no modelo relacional, regra da sub linguagem abrangente, regra da atualização de visões, independência dos dados físicos, independência lógica de dados, independência de integridade, independência de distribuição e regra da não-subversão.

Implementação e modelagem: parte I

Qualquer estratégia de implementação de banco de dados tem um processo inicial de compensação dos requisitos que o software tem que atender dentro do ciclo de engenharia de requisitos.

O processo de modelagem de dados de sistemas transacionais tem por objetivos três tipos de modelagens: a primeira providênci a é representar de uma maneira inequívoca a construção de um esquema conceitual dos dados, sendo a modelagem conceitual a necessidade de compreender os requisitos para identificar quais dados armazenar.

23:13



“Pensar que sistemas relacionais ‘já eram’ e que agora é NoSQL é, no mínimo, ingenuidade.
”

30:48

32:16



“Modelar é uma maneira de representarmos a compreensão dos requisitos.
”

Implementação e modelagem: parte II

No modelo conceitual, temos a visão sobre o que será armazenado no banco de dados; no lógico, pensamos em como serão armazenados os dados no banco de dados, identificando as melhores estruturas lógicas de dados ideais para a implementação. Para uma implementação de alta complexidade de um sistema robusto, não se dispensa modelagem física, na qual se definem as estruturas lógicas: implementa-se o banco de dados, criando suas estruturas de dados.

Os modelos de dados são utilizados para descrever a estrutura de um banco de dados. São empregadas abstrações para representar os objetos de realidade que serão modelados, como suas características e associações.

44:44



PALAVRA-CHAVE

James Rumbaugh: é um cientista da computação e metodologista do modelo de desenvolvimento de software orientado a objetos, mais conhecido por criar o Object Modeling Technique e a Unified Modeling Language.

PALAVRAS-CHAVE

Ivar Hjalmar Jacobson: é um cientista da computação sueco que concluiu seu mestrado em engenharia eletrônica no Chalmers Institute of Technology de Gotemburgo em 1962 e um Ph.D. no Royal Institute of Technology de Estocolmo em 1985.



45:03

PALAVRAS-CHAVE

Astah: é uma ferramenta de modelagem UML criada pela empresa japonesa Change Vision. Recebeu o prêmio “Software Product Of The Year 2006”, estabelecido pela Information-Technology Promotion Agency no Japão.



45:23

PALAVRA-CHAVE

Grady Booch: informático e autor do livro *Software Engineering with Ada*, que lançou as raízes do projeto orientado a objetos. Esse trabalho evoluiu para uma metodologia de desenvolvimento de sistemas orientados a objetos publicada em seu livro *Object-Oriented Analysis and Design with Applications*.



46:34

AULA 1 • PARTE 2

Modelagem conceitual: parte II

Entidade no modelo entidade/relacionamento e classe no modelo orientado a objetos têm por objetivo representar coisas da realidade que nós estamos modelando. Normalmente, enquadrados dentro das categorias pessoas, objetos inanimados, documentos, locais e eventos, e são compostas por atributos, que representam características das entidades/classes. A instância/objeto é como se denomina cada ocorrência de uma entidade/classe que representa um único objeto da realidade.



00:17



12:01

Modelagem conceitual: parte I

O objetivo da modelagem conceitual é a representação dos requisitos, necessidades dos usuários (compreensão do que está sendo mostrado). Aqui se apresentam os relacionamentos ou associações possíveis entre instâncias/objetos da realidade. Dentro desses relacionamentos, existem tipos relacionados à cardinalidade: os não identificadores e os identificadores. O não identificador é composto somente pelo seu atributo, enquanto o identificador é composto pela classe e pelo atributo.

13:30



“ Tentem, na medida do possível, modelar aquilo que vocês têm certeza de que precisa compor o modelo de dados. **”**

Relacionamentos

Dentre os relacionamentos temos os tipos:

1 para N: tem uma categorização especial, podendo ser não identificadores ou identificadores. O objeto pode ter uma existência associada ou não com outro objeto;

N para N: são representados como uma ligação entre os relacionamentos. Podem possuir atributos, com cada instância N para N representando uma associação entre pedido e produto.



17:26

EXERCÍCIO DE FIXAÇÃO

É o relacionamento que possui dois tipos, identificadores e não identificadores:



Entidades/classes associativas

As entidades/classes associativas representam as associações entre instâncias de relacionamentos N para N, com outras entidades/classes.

Outra abstração são os relacionamentos unários, aqueles que envolvem somente uma entidade/classe. Ou seja, as instâncias de uma entidade estabelecem relacionamentos com instâncias da mesma entidade.



34:00

Hierarquias de generalização

As hierarquias de especialização permitem reunir em classes/entidades mais genéricas os atributos e os relacionamentos mais comuns a várias entidades/classes. Usa-se de exemplo pessoa jurídica e pessoa física, pois ambas são clientes. Quando se juntam os atributos de uma pessoa física e de uma pessoa jurídica, transformam-se os atributos das duas pessoas na classe CLIENTE, fazendo uma generalização. As especializações podem ser totais ou parciais, e é dita exclusiva quando não se pode estar em duas classes especializadas ao mesmo tempo.



43:51



44:16

PALAVRAS-CHAVE

Stakeholder: um dos termos utilizados referente às partes interessadas que devem estar de acordo com as práticas de governança corporativa executadas pela empresa.

EXERCÍCIO DE FIXAÇÃO



Qual classe não é capaz de capturar todos os perfis de determinada categoria?

AULA 1 • PARTE 3

Modelo lógico relacional

O modelo lógico relacional tem por objetivo definir as estruturas de dados que serão utilizadas para a implementação dos requisitos, criando um modelo conceitual de dados, como modelo entidade-relacionamento ou modelo orientado a objeto.

As tabelas são os elementos básicos para prover persistência a dados em SGBDs relacionais. Nelas, são possíveis as definições de valores padrão, regras de validação de valores, restrição de nulidade, chaves primárias, chaves alternativas e chaves estrangeiras.



00:17



EXERCÍCIO DE FIXAÇÃO

Refere-se à definição de restrição de integridade:

EXERCÍCIO DE FIXAÇÃO

Estabelecem a obrigatoriedade de que uma ou mais colunas de uma tabela sejam únicas em conjunto:



16:24



Surrogate keys

Para escolher os valores das chaves primárias, o recomendável é a utilização de chaves artificiais, as surrogate keys. O professor Eduardo apresenta três opções de criação de surrogate keys.

19:55



Normalização de dados

A normalização de dados é o processo pelo qual cria-se um modelo relacional de dados pela aplicação de formas normais a dados brutos, ditos não normalizados, visando obter dados livres de efeitos colaterais de atualizações. É baseada em um conceito de forma normal, uma regra a ser obedecida a uma tabela que seja considerada “bem protegida”.

Primeiro, devemos identificar quais atributos identificam o fato, e então remover a multi valoração, mantendo o vínculo com o identificador original. Eliminam-se as dependências funcionais parciais, mantendo o vínculo com o identificador.

“ Normalização de dados é um conceito fundamental em modelagem de dados relacionais. **”**

Teorema CAP

Os sistemas de gerência de banco de dados devem garantir a consistência, a disponibilidade e a tolerância de partição ou particionamento. Com base nesses três requisitos, em 1988, Eric Brewer propôs o Teorema CAP, que diz que é impossível em um ambiente conseguir conciliar os três requisitos simultaneamente. Sempre que tivermos a escolha dos dois requisitos, o terceiro deve ser relativizado. O único jeito dos três trabalharem e não ter erro, é um único nó.



30:00

31:13



“ Sempre que eu fizer uma consulta, em qualquer um dos nós de um SGBD, tenho que receber uma resposta. **”**

32:44



PALAVRA-CHAVE

Eric Brewer: é professor emérito de ciência da computação na Universidade da Califórnia, Berkeley, e vice-presidente de infraestrutura do Google. Seus interesses de pesquisa incluem sistemas operacionais e computação distribuída.

Modelos de transações de SGBDs

O conceito de transações de dados foi iniciado na década de 70, com o advento dos bancos de dados relacionais de modelo com requisitos ACID (Atomicidade, Consistência, Isolamento, Durabilidade).

Esse conceito foi evoluindo com a chegada dos bancos de dados não relacionais, passando a relativizar, em função do Teorema CAP, a capacidade de garantir os requisitos ACID em ambientes com distribuição, passando a levar em consideração o modelo de transações de requisitos BASE (Basicamente disponível, Soft state, Eventualmente disponível).

41:46



43:53



“*Não existe equivalência entre a classificação do Teorema CAP e a classificação ACID e BASE.*”

EXERCÍCIO DE FIXAÇÃO

“Uma transação processada nunca colocará em risco a integridade estrutural do banco de dados”. A afirmação se refere ao conceito de:

48:44



PALAVRA-CHAVE

Two-phase commit protocol: é um algoritmo distribuído que coordena todos os processos que participam de uma transação atômica distribuída para confirmar ou abortar a transação.

AULA 1 • PARTE 4

Exemplo prático: parte I

O professor Eduardo Pereira de Arruda realiza um exercício prático e explica o MongoDB conceitualmente, exemplificando um cadastro básico de um cliente, trazendo como seria a divisão de tabelas através de um modelo conceitual no Astah. Demonstra como fazer a normalização pensando na maneira como os dados serão consumidos.



00:17

15:41



Exemplo prático: parte II

Seguindo o exemplo prático, o professor Eduardo Pereira de Arruda apresenta as duas alternativas, modeladas anteriormente, implementadas no BSON. Destaca-se que uma redundância controlada pode garantir melhor desempenho.

Exemplo prático: parte III

O professor demonstra a especialização exclusiva, praticando os conceitos vistos em aula, destaca que o processo é cíclico. Demonstra ainda, na prática, os tipos de relacionamentos.



30:42

AULA 2 • PARTE 1

Introdução

O professor Vinicius Kroth introduz o que será visto em aula: exemplos práticos, teoria e prática com banco de dados, e uma visão sobre as arquiteturas dos bancos. Traz ainda os pré-requisitos de instalação para rodar os softwares e retoma algumas palavras chaves que são necessárias dentro de um banco de dados.

“ Entender com o que se está trabalhando ajuda muito a escolher o banco de dados. **”**



02:32

EXERCÍCIO DE FIXAÇÃO

Refere-se a um agrupamento de unidades computacional:



11:08



Não deixem uma experiência com banco de dados relacionais mal aplicada, mal desenhada, mal modelada, influenciar a opinião de vocês com esses bancos.



13:11



NoSQL

Os casos de usos estão cada vez mais complexos e os clusters, cada vez mais comuns. Além disso, temos uma maior quantidade de dados, e os NoSQL são sistemas que aguentam quantidades de dados extremas.



14:54

Um dos pontos mais falados do NoSQL é a flexibilidade na estrutura de dados: alguns bancos de dados precisam de uma pré estrutura para a alocação de dados no mesmo, já outros (como mongoDB e NoSQL) não precisam de uma pré estrutura definida, dando uma flexibilidade e uma naturalidade entre dados não iguais.

PALAVRAS-CHAVE

Amazon Aurora: é um serviço de banco de dados relacional desenvolvido e oferecido pela Amazon Web Services. Está disponível como parte do Amazon Relational Database Service.

“A gente tem uma flexibilidade muito grande nos bancos de hoje em dia.**”**

“Perfeito em todos os aspectos é uma mentira.**”**



19:11

22:13



MongoDB

Disponibilizado em 2009, o mongoDB é um banco que teve grande transformação ao longo dos anos. Seu sistema atual conta com diversos privilégios para quem o usa, tendo um esquema de dados flexível suportando uma documentação extensiva, ainda oferecendo uma linguagem baseada em Javascript, sendo nativamente escalável, entre muitos outros benefícios. Sua unidade básica é chamada de documento. Os docs são armazenados no formato BSON, sendo que todo doc tem um identificador único com formatos variados. Os documentos são agrupados em coleções, e o server é nativamente clusterizado, entre muitos outros conceitos e benefícios.



26:21

EXERCÍCIO DE FIXAÇÃO

Como se chama um banco de dados sem esquema?



37:21



Escalabilidade dos servidores

Escalabilidade vertical é onde melhoramos os equipamentos que já temos e expandimos; na horizontal, duplicamos o equipamento, com melhores hardwares e espaços, ou a mesma quantidade de armazenamento e CPU.

Em escalabilidades horizontais, temos o tipo sharding (uma divisão do servidor em partes), o tipo mestre escravo (onde uma parte fica responsável pela leitura e a outra, pela escrita) e o tipo peer-to-peer (que visa uma divisão de gravar/ler todos os dados, além de replicar seus dados com mais nodos).

PALAVRAS-CHAVE

Hash table: é uma estrutura de dados especial que associa chaves de pesquisa a valores, com o objetivo de, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado.



41:27

41:47



PALAVRA-CHAVE

PALAVRAS-CHAVE

Failover: significa tolerância a falhas. Quando um sistema, servidor ou outro componente de hardware ou software fica indisponível, um componente secundário assume operações sem que haja interrupção nos serviços.



45:11

Load balance: soluções para especializar em pequenos grupos sobre os quais se faz um balanceamento de carga: utilização do CPU, de armazenamento ou de rede.

AULA 2 • PARTE 2

EXERCÍCIO DE FIXAÇÃO

A escalabilidade do MongoDB começa com a aplicação do cliente sendo lida e escrita pelo:

00:17



Escalabilidade: prós e contras

São os prós e contras das escalabilidades horizontais:

- sharding: divide a carga de escrita e leitura, porém, se um shard cai, aquela fração de dados se torna indisponível;
- replicação mestre/escravo: melhora a capacidade de leitura e aumenta a disponibilidade dos dados, porém, pode gerar problemas de consistência ou disponibilidade e não melhora a capacidade de escrita, além de criar um SPOF;
- peer-to-peer: melhora a capacidade de leitura e escrita, aumenta a disponibilidade dos dados, porém, pode gerar problemas de consistência ou disponibilidade, além de custo elevado.

11:29



Desvantagens dos tradeoffs

O tradeoff não possui suporte para transações ACID e JOINS, tem um limite de 100mb de memória, tem duplicação de dados - o que pode se tornar um problema - , não serve para todos os casos, a consistência não é o ponto forte desse banco, tem limite para armazenar documento físico e sua utilização de memória para armazenamento é grande.

“ É uma ótima prática a gente pensar muito bem em como vai ser o nosso documento.”

Exemplo prático: parte I

O professor Vinícius Kroth traz um exemplo prático demonstrando: modelagem de dados, introdução à linguagem, busca simples, busca complexa, interação entre coleções e agregações. Destaca que é importante pensar em como será armazenado antes da modelagem. Um diagrama pode auxiliar na orientação e na organização da equipe.

21:08



“ Não confunda modelagem de dados com esquema de armazenamento de dados.”

16:36



Exemplo prático: parte II



34:33

Seguindo o exemplo prático, o professor Vinícius Kroth demonstra como fazer uma inserção e agregação de dados. Em seguida, para dar um maior valor aos dados, realiza uma análise e projeção.

AULA 2 • PARTE 3

PALAVRAS-CHAVE

Consistência eventual: é um modelo de consistência utilizado na computação distribuída com a finalidade de se atingir a alta disponibilidade, garantindo-se informalmente que, se nenhuma nova atualização de um dado for realizada, posteriormente, todos os acessos a esse dado retornarão o último valor atualizado.

00:16



Cassandra DB: parte I

O Cassandra DB é um banco de dados distribuído altamente escalável. Ele tem uma performance excelente de escrita de dados; sua nuvem nativa permite replicação em diversas zonas, tendo uma arquitetura masterless e evitando pontos únicos de falhas; é altamente disponível e tolerante a falhas, sendo facilmente e nativamente horizontalmente escalável. Tem como principais conceitos um suporte em largas escritas, utilizando CQL; trabalha eventualmente com consistência eventual; trabalha com chave primária; organiza os dados em tabelas; sua arquitetura é em anel e divide os dados nativamente pelos nodos.



04:24

EXERCÍCIO DE FIXAÇÃO



Qual a estrutura da arquitetura do Cassandra DB?

“Quanto mais partições eu vou, mais elementos eu adiciono na minha complexidade e mais latência.**”**

Cassandra DB: parte III

As estratégias de compactação de dados que podem ser utilizadas são: STCS (excelente em cenários onde há muita escrita), LCS (recomendada em cenários com muita escrita e atualizações) e DTCS (apropriada para dados com uma coleção de observações feitas sequencialmente ao longo do tempo).

O modelo de escrita do Cassandra DB é um pouco diferente, ele faz um combinado de dados do memtable e SSTable e retorna os dados, tornando o banco de dados indicado para escrita pesada. As consultas no Cassandra são muito dependentes de como os dados são organizados.

Casos de uso

São os casos de uso do Cassandra DB:

- e-commerce e gestão de inventário;
- eventos time-series;
- serviços de pagamentos / financeiros;
- distribuição e armazenamentos de conteúdo.

Como toda tecnologia, o Cassandra também tem os seus tradeoffs: não suporta transações ACID; duplicação de dados; leituras mais lentas que escrita; não conta com subqueries; agregações ou joins nativos; consistência não é um ponto forte e não serve todos os casos.

13:00



Cassandra DB: parte II

São conceitos importantes do banco de dados Cassandra DB:

- Keyspace: similar ao database dos bancos relacionais;
- Tabela: similares ao SQL;
- Primary key: um identificador único de uma tabela que pode ser composto;
- Partition key: direciona os dados para sua partição correspondente;
- Clustering key: uma chave derivada da primary key, que ordena os dados dentro de uma repartição.
- Na arquitetura do Cassandra DB os dados não são deletados completamente, os chamados “tombstones” afetam a performance da queries, sendo mais interessante utilizar o soft deletes, onde os dados são eventualmente removidos.

15:43



21:39



EXERCÍCIO DE FIXAÇÃO

Assinale a alternativa que reflete uma consequência do aumento da disponibilidade de dados no Cassandra DB:

32:43



AULA 2 • PARTE 4

Exemplo prático: parte II

Seguindo o exemplo prático, o professor Vinícius Kroth realiza consultas simples no banco de dados. Em seguida, altera a tabela e a forma que os dados são inseridos para resolver o problema de modelagem padrão.

00:17



Exemplo prático: parte I

O professor traz um exemplo prático demonstrando uma introdução à linguagem; como isolar pedidos usando UDT; como remodelar os pedidos; como ver a utilização das keys no Cassandra e dividir os dados em múltiplas tabelas.

Exemplo prático: parte III

O professor Vinícius Kroth destaca que a ordem de criação das chaves importa, e demonstra como acrescentar uma busca de forma eficiente. Relembra a importância de pensar nas chaves para uma boa performance e atender aos requisitos.

19:37



“ As tabelas são derivadas das queues. **”**

25:28

AULA 3 • PARTE 1

Redis: principais conceitos

No Redis, todo valor salvo é atrelado a uma chave única, visto que sua arquitetura é em single-thread, fazendo com que os valores possam ter formatos/estruturas diferentes. Apesar de não oferecer paralelismo, tem suporte a concorrência. Além disso, todos os dados são mantidos em memória (RAM).

04:23



Redis: por que usar?

O Redis é uma tecnologia de banco de chave-valor open source, porém tem algumas features que só funcionam na versão premium. Com um excelente throughput de escrita e leitura, tem um esquema de dados flexível com uma linguagem simples, fazendo com que haja uma integração com muitos clientes. É facilmente escalável e suporta adição de módulos com funcionalidades específicas, fazendo com que o Redis seja visto como um banco de cache.

11:24



PALAVRAS-CHAVE

Paralelismo: quando as tarefas de um programa são executadas em simultâneo em mais de um processador.

12:03



PALAVRA-CHAVE

Single-thread: é uma aplicação que executa apenas uma thread por vez.

18:17



Modelos de persistência

Apesar de ser um banco de cache, o Redis tem persistência, com o objetivo de recuperar o servidor caso ele caia. São os tipos de persistência:

- RDB (Redis database): realiza snapshots do banco de dados, de tempos em tempos, salvando o atual. Comum em casos onde o Redis é um banco secundário;
- AOF (Append Only File): cria um arquivo log que salva todos os comandos repetitivamente quando o banco é reinicializado. Comum em casos onde o Redis é o banco principal/fonte da verdade;
- Sem persistência: desabilita a persistência por completo. Comum em casos onde o Redis é utilizado por caching.

20:47



EXERCÍCIO DE FIXAÇÃO

Quando os processos disputam um ou mais elementos computacionais denominamos:

Redis: casos de uso

O Redis é muito utilizado para: cache de informações frequentemente acessadas, armazenamento de informação de sessão de usuário, sistema de mensagens (chats), placares de jogos e análise de dados em tempo real (Redis Search). 70% a 80% das aplicações que usam o Redis são só para cache.

33:28



EXERCÍCIO DE FIXAÇÃO

Qual módulo do Redis dá a funcionalidade de indexar e realizar buscas por elementos?

38:16



PALAVRA-CHAVE

Pub/sub: é um serviço de mensagens assíncrono e escalonável que separa os serviços que produzem mensagens dos serviços que processam essas mensagens. Ele permite que os serviços se comuniquem de forma assíncrona, com latências de 100 milissegundos.

Redis: desvantagens

O Redis tem algumas desvantagens: o single-thread pode gerar gargalo, custo de armazenar os dados em memória; a natureza é efêmera, podendo gerar downtime; há limite físico no armazenamento de cada valor; a funcionalidade completa da aplicação é somente para as versões premium que, sem a utilização dos módulos, só é eficiente para acessar os dados pela chave.

40:23



EXERCÍCIO DE FIXAÇÃO

A natureza efêmera do Redis pode gerar:

AULA 3 • PARTE 2

Exemplo prático: parte I

O professor Vinicius Kroth apresenta de forma prática a utilização do Redis para cachejar os dados, junto com o Redis Search para criar um índice, realizando operações de FTS.

00:17



03:01



PALAVRA-CHAVE

Redis Stacking: uma extensão do Redis que adiciona modelos de dados modernos e mecanismos de processamento para fornecer uma experiência completa ao desenvolvedor.

Exemplo prático: parte II

Seguindo o exemplo, o professor Vinicius Kroth demonstra como realizar uma busca no Redis levando em consideração os documentos, trabalhando com o banco de dados um pouco fora do padrão, utilizando o Redis Search.

09:39

PALAVRAS-CHAVE

Janela de inconsistência: um período entre a atualização e o momento em que é garantido que qualquer observador sempre vai olhar o valor atualizado.

18:13

Exemplo prático: parte III

O professor Vinicius Kroth destaca que quando um índice já está criado e é feito um set de qualquer objeto dentro daquele índice, significa que aquele documento já foi criado e indexado. Demonstra ainda como dar suporte à agregação.

19:01

AULA 3 • PARTE 3

FUNDAMENTO

Bancos orientados a grafos

É uma plataforma especializada e de uso único para criar e manipular grafos. Os grafos contêm nós, arestas e propriedades, todos usados para representar e armazenar dados de uma forma que os bancos de dados relacionais não estão equipados a fazer.

00:16

Neo4j

O Neo4j faz parte dos bancos orientados a grafos. Tem como principal característica facilitar a busca de caminhos com relacionamentos. Com um suporte a transações ACID, ele provém do protocolo de comunicação REST, tendo uma base de dados flexível, uma performance boa em situações de conexões/relacionamentos entre dados, e, em comparação com outros bancos, sua consulta é mais eficiente.



00:16

EXERCÍCIO DE FIXAÇÃO

Conceito que traz as informações de cada nodo:

Neo4j: benchmarks oficiais

Imaginando um cenário onde precisamos buscar relações que vão de dois a cinco níveis de uma pessoa nas redes sociais, levando em consideração uma média de 500 pessoas em cada nível de relacionamento, conforme procuramos amigos de amigos em buscas, o Neo4j pode ser até 100 vezes mais rápido que o MySQL em uma consulta com uma carga imensa de dados.



08:50

“ As pessoas estão separadas por um certo nível de conexão. **”**



09:40

Neo4j: escalonamento e usos

No Neo4j, a replicação de nodos por disponibilidade é mais eficiente, tendo as informações passadas por uma fábrica. Essa estrutura pode ter oscilações, gargalos e pontos de falhas.

O banco de dados Neo4j pode ser utilizado em: redes sociais, criando relacionamentos entre nodos; detecção de fraude; recomendação de conteúdo; controle e gestão de identidade; análise de riscos; mapas; e coordenação de rotas.



18:01

“ Pensem em como vocês querem acessar os dados antes de modelarem o nodo. **”**



19:51

29:55

Neo4j: desvantagens

O Neo4j é para caso de usos específicos, não de propósito geral; há uma falta de suporte para operações de agregação; a comunidade é relativamente pequena; não apropriado para escritas massivas; é um banco excelente para relações, mas o uso errado com um estudo raso do banco pode gerar em prejuízo enorme.



EXERCÍCIO DE FIXAÇÃO

NÃO é um uso indicado do banco de dados Neo4j:



AULA 3 • PARTE 4

Exemplo prático: parte II

Seguindo o exemplo, o professor Vinicius Kroth muda um pouco a dinâmica, especificando os relacionamentos de forma direta. Em seguida, muda a cardinalidade da relação, para acessos mais frequentes, e determina o menor caminho possível entre objetos.

00:17



10:43

Exemplo prático: parte I

O professor Vinicius Kroth traz um exemplo prático pronto do Neo4j, apresentando uma introdução à linguagem CYPHER, visualização de grafos, operações em datasets de exemplo. O código pode ser acessado em: <https://Neo4j.com/developer/example-data/>

20:30



Dicas

O professor Vinicius Kroth revisa o conteúdo e aponta que não adianta moldar um banco sem pensar em como os dados vão ser acessados e manipulados. Vinicius sugere que evitemos situações onde se tenha que criar um esquema prévio, porque esse não é o objetivo do banco relacional. Consistência é um bom ponto para desacoplar o sistema e reduzir a carga do sistema. Tenha noção do quão frequente os dados serão acessados. Busque custear um banco mais barato, mas pensando em quanto tempo aquele dado irá ficar armazenado.

“ Existem cenários onde eu preciso que meus dados sejam muito seguros.



PALAVRAS-CHAVE

Two-phase commit: é um algoritmo distribuído que coordena todos os processos que participam de uma transação atômica distribuída para confirmar ou abortar a transação.

32:00



Outras alternativas

Se a proposta não for utilizar um banco em específico, pode-se utilizar bancos que não são tão otimizados para alguns aspectos, mas que conseguem manter tudo em ordem. Lembrando que manter bancos diferentes não é tão saudável para os dados, por causa da consistência que precisa ser mantida.

Um banco relacional deve resolver a maioria dos problemas. Se o requisito dos dados não for muito específico, vá para um banco de dados relacional, pois é uma escolha mais segura.

“ Um banco relacional deve resolver a maioria dos problemas. **”**

35:31



38:50



LEITURAS INDICADAS

O professor Vinicius Kroth sugere os seguintes livros para consulta:

NoSQL Essencial;

Designing Data-Intensive Application;

Philosophy of Software Design.

Resumo da disciplina

Veja, nesta página, um resumo dos principais conceitos vistos ao longo da disciplina.

AULA 1

A finalidade principal de um banco de dados é garantir a persistência.



Normalização de dados é um conceito fundamental em modelagem de dados relacionais.

Não existe equivalência entre a classificação do Teorema CAP e a classificação ACID e BASE.



AULA 2

Não confunda modelagem de dados com esquema de armazenamento de dados.



Entender com o que se está trabalhando ajuda muito a escolher o banco de dados.



As tabelas são derivadas das queues.



AULA 3

Demonstração dos bancos de dados sugeridos.



Pensar em como iremos acessar nossos dados antes de modelar o nodo é essencial.



Um banco relacional deve resolver a maioria dos problemas.



Avaliação

Veja as instruções para realizar a avaliação da disciplina.

Já está disponível o teste online da disciplina. O prazo para realização é de **dois meses a partir da data de lançamento das aulas**.

Lembre-se que cada disciplina possui uma avaliação online.
A nota mínima para aprovação é 6.

Fique tranquilo! Caso você perca o prazo do teste online, ficará aberto o teste de recuperação, que pode ser realizado até o final do seu curso. A única diferença é que a nota máxima atribuída na recuperação é 8.

