Relações com banco de dados, embora seja um tema um pouco negligenciado por algumas pessoas, entender que é muito importante, pois é importante no dia a dia de todo programador entender sua relação com o banco, o tipo de banco de dados no processo. escolha o melhor para o desenvolvimento de sua aplicação.

Um banco de dados relacional é um tipo de banco de dados, existem outros tipos de bancos de dados, como bancos de dados não relacionais, bancos de dados relacionais são baseados no tipo de organização relacional para exibir e organizar os dados em tabelas. Nesse sentido, podemos acessar diversos tipos de dados, como dados financeiros de empresas, informações de pacientes em hospitais, estudantes e universidades, entre outros. Quando nos referimos ao modelo desse relacionamento, nesse tipo de banco de dados, cada linha da tabela representará um identificador único, que chamaremos de chave. É útil definir outros atributos que fazem parte de um modelo de banco de dados relacional, como colunas, que contêm atributos de cada dado e também possuem valores.

Para construir um banco de dados, temos uma variedade de objetivos que fazem parte dele. Entre eles:

Tabela: É responsável por armazenar dados, como nomes e anos de pessoas que estudam ciência da computação em uma determinada universidade. A tabela também pode ser chamada de relação porque um banco de dados relacional pode ter uma ou mais relações. Coluna: É responsável por armazenar apenas um tipo de dado, que pode ou não ser nulo. É importante lembrar que esta pode ser uma coluna não chave, ou seja, o tipo de dado pode ser repetido em outras colunas. Tupla: É responsável por representar cada registro de dado dado. O registro é identificado pela chave primária, pois não é possível ter um registro duplicado. Relacionamento: Preste atenção nessa parte, relacionamento é diferente de relacionamento. Como o Relacionamento representará a tabela, o relacionamento será um elo entre eles, pois estão vinculados pela chave primária e pela chave estrangeira. Chave primária: Do inglês "chave primária" chamada "pk", uma chave primária é um atributo (ou coluna) que identifica um registro exclusivo. Por exemplo, temos o CPF de um aluno. Chave estrangeira: também chamada de "chave estrangeira" ou "fk", é um atributo que define como as tabelas são unidas. Em outras palavras, fk está conectado a pk. Por exemplo, podemos definir uma tabela "gosto musical", com chave estrangeira, que criará uma relação com a chave primária na lista de alunos.

Sendo que os relacionamentos entre bancos de dados são;

Um para um: quando nos referimos a esse tipo de relacionamento, estamos dizendo que um registro em uma tabela está ligado a apenas um registro em outra tabela/relação. Para exemplificar a situação, podemos utilizar ainda estudantes do curso de Ciência da Computação, cada estudante tem um ID exclusivo e cada ID é atribuído a apenas uma pessoa.  
Um para muitos: aqui estamos falando que a tabela de chave primária pode se relacionar com nenhum, com um ou muitos registros da tabela que está sendo associada.  
Muitos para muitos: dado as definições anteriores, esse tipo de relacionamento pode se relacionar a nenhum ou a vários registros em outra tabela. Aqui, nesse tipo de relacionamento é necessário uma terceira relação/tabela.

Vale destacar que os bancos de dados relacionais podem estar acoplados para diferentes fins, entre eles:  
 Documentação: refere-se a quando esse sistema de banco muda, logo a documentação também deve ser alterada, Scripts para migração de dados: refere-se às mudanças de script que requerem migração, O código fonte da aplicação: refere-se às alterações ocorridas no código fonte da aplicação que quando alterada o esquema também precisa ser modificado, O código fonte para extração de dados: refere-se a scripts e/ou programas de extração de dados, muitas vezes responsáveis por gerar arquivos em XML ou por carregar dados em outro banco.

Ok, mas e sobre os procedimentos de rastreamento em banco de dados sem que interfira na usabilidade da pessoa desenvolvedora durante o seu acesso ao banco de dados? Tudo isso é possível graças ao processo de encapsulamento.   
 O encapsulamento é um meio para a implementação de alterações de modo “invisível”, no qual é possível fazer modificações dentro do banco sem modificar outros sistemas — diferente do acoplamento, em que qualquer alteração precisaria de um conjunto de alterações dentro do banco de dados.

Além disso, outra coisa interessante é o fato de o DBA poder trabalhar sem nenhuma restrição em seus planos e uso do banco sem afetar a implementação de muitas outras regras desenvolvidas por outros grupos. Portanto, os programadores podem se concentrar apenas em resolver seus problemas de implementação sem se concentrar na segurança dos dados. Ou seja, tudo isso promove maior independência para os profissionais. Infelizmente, nem sempre é possível não usar uma conexão, mas o plano de seguro é muito útil quando adequado ao projeto em questão. Portanto, também existem desvantagens como incompatibilidade entre plataformas/sistemas. No entanto, embora possamos reduzir a conexão, teremos uma vantagem, pois está associada a uma redução de trabalho e, portanto, de custos, nesse sentido, é necessário ter um banco de dados salvo que atenda às necessidades do seu problema. Aqui estão alguns pontos-chave que podem ajudá-lo no processo de seleção:

Funcionalidade e a importância da confiança: Já mencionamos neste mesmo artigo a importância de ter um sistema de segurança, para que a confiança e segurança de nossos dados se torne um fator importante nesse processo de tomada de decisão. Além disso, o desempenho deste sistema precisará suportar uma grande quantidade de dados para que funcione de forma rápida e eficiente? Portanto, este será um fator importante nesta decisão.

Funcionamento simultâneo: Se um grande número de usuários utiliza este banco e se conecta a diferentes aplicativos, é importante saber o quanto este sistema pode suportar para não prejudicar o desempenho do seu produto.

Escalabilidade: quando nos referimos a escalabilidade precisamos entender sobre a base de dados que estamos lidando, caso seja um volume muito grande, existirá, certamente um produto que nos auxilie melhor nessa questão.

OK, mas quais são as principais características do SGBD?

Em um SGBD, existe uma linguagem para gerenciar a quantidade atual de dados. Entre eles, há CRUD, sigla em inglês para CREATE, READ, UPDATE e DELETE, e um resumo das vantagens e desvantagens;

Vantagens: Quando falamos de bancos de dados relacionais, estamos falando de maneiras de organizar nossas informações de forma eficiente. Então, já imaginou ter que gerenciar múltiplas fontes de informação sem a ajuda dessa tecnologia? Parece impossível, não é? Bom, antigamente funcionava assim, mas hoje, graças aos bancos de dados relacionais, podemos criar esses dados de forma estruturada, melhorando assim nosso trabalho. Contras: No entanto, você já pensou no trabalho envolvido na criação desses sistemas? Portanto, é aí que surge um dos problemas mais sérios, pois criar um banco de dados relacional é complexo. Nesse sentido, a criação de relacionamentos, a definição de chaves e conexões entre tabelas dificulta esse processo.

Para desenvolver o trabalho, vamos dividi-lo em três partes, que são: trabalho conceitual, trabalho racional e trabalho físico.

Plano conceitual: Refere-se à fase de planejamento e às expectativas e necessidades do cliente com o desenvolvedor, durante a qual é criada a grande visão do projeto. Trabalho lógico: refere-se ao definir os requisitos e conceitos deste trabalho para sua organização, de tabelas, metadados de coluna, etc. sendo esquematizado.

Trabalho físico: refere-se a quando são feitas as especificações técnicas para implementação do banco de dados, que são definidas por termos como tipos de armazenamento, scripts e outros.

NoSQL

Os bancos de dados NoSQL (não relacionais) nasceram da necessidade de usar soluções mais simples à medida que os negócios crescem. Os bancos de dados NoSQL (ou não relacionais) usam um padrão de segurança diferente do SQL. A grande vantagem dessa tecnologia é a possibilidade de criar operações de negócios de forma mais simples e econômica do que em bancos de dados relacionais, e o NoSQL proporciona melhor desempenho para gerenciamento de dados estruturados, pois não há necessidade de combinar dados e configuração de tabelas. usando informações.

Carlo Strozzi usou o termo pela primeira vez em 1998, quando falava sobre um banco de dados não relacional. Com o aumento da utilização da Internet e das soluções digitais, torna-se cada vez mais importante encontrar uma forma inteligente de gerir as bases de dados empresariais. A questão foi retomada em 2006, quando a empresa Google publicou um artigo sobre segurança de dados. Em 2009, o patrocinador Rackspace organizou um evento para discutir bancos de dados de código aberto e mencionou o NoSQL novamente. Nessa época, muitas tecnologias com bancos de dados não relacionais começaram a aparecer e o tema se popularizou. As principais bases de dados não relacionais utilizadas em marketing são;

Diga isso de novo; Redis é o banco de dados de cadeia de valor NoSQL mais usado no mundo. Ele combina valores e chaves com sua estrutura, facilitando o armazenamento e a recuperação dos dados. Portanto, os desenvolvedores costumam usá-lo.

memcached; Bancos de dados não relacionais também implementam armazenamento baseado em valor e usam memória cache distribuída. É frequentemente usado para criar sites dinâmicos, pois acelera o carregamento das páginas e reduz a busca por dados de diferentes fontes. Cassandra; Este banco de dados NoSQL foi desenvolvido pelo Facebook. Ele usa um banco de dados descentralizado, onde os dados são armazenados em vários data centers. Ele é configurado para clusters e fornece baixa latência em suas atualizações.

Hbase; Hbase é um banco de dados que usa uma estrutura de linha e coluna para armazenar informações. É usado em várias plataformas, como LinkedIn, Facebook e Spotify

Amazon DynamoDB; É um banco de dados em nuvem NoSQL fornecido pela Amazon Web Services. Possui baixa latência, é rápido e flexível, tornando-o um modelo ideal para aplicativos móveis, jogos web e soluções de IoT. Ele também oferece alto desempenho e escalabilidade automática, recursos importantes para empresas que desejam crescer com eficiência.

Neo4j; O Neo4j é um banco de dados não relacional baseado em gráficos (interfaces e nós). É uma implementação de código aberto e pode ser útil para casos de mineração de dados e reconhecimento de padrões.

MongoDB; Também é um bom banco de dados de código aberto. É suportado por vários sistemas operacionais e caracterizado por sua estrutura de documentos.

Como tal, ele armazena todas as informações relevantes em um documento e usa métodos avançados de classificação e filtragem. O MongoDB suporta uma variedade de frameworks e linguagens, incluindo Java, JavaScript, PHP, Python e Ruby.

Vantagens e desvantagens;

Vantagens;

Como não está interessada em uma estrutura das inter-relações da tabela, em relação aos dados de dados, ela se torna muito flexível. Além disso, é sempre fácil adicionar novos dados, sem que a estrutura lógica da tabela faça alterações. O motor tem uma boa relação com Big Data e suas peculiaridades. Portanto, há maior capacidade para suportar grandes dados. Portanto, a manutenção é realmente automática, eliminando a necessidade de funcionários.

Desvantagem;

No caso de trabalhar com grandes dados dos usuários do site, por exemplo, o armazenamento de dados não relacionais não é eficaz. Nesse sentido, é porque sua estrutura não foi projetada para esse tipo de informação. Se for necessário contar o número de comentários em uma página web, por exemplo, ou o número de posts, não é possível fazê-lo, utilizando NoSQL. Banco de dados relacional x não relacional: a diferença e o que é melhor para você Já explicamos em todo o texto o que é um banco de dados relacional, ou seja, ele é responsável por armazenar e organizar os dados que os conectam entre si. E, agora, vamos falar sobre o que é um banco de dados não relacional. Nesse sentido, quando nos referimos a um banco de dados não relacional, estamos falando de qualquer banco de dados que não segue o padrão RDBMS, portanto, são conhecidos como parte do sistema NoSQL. Entre eles estão os mais populares, como MongoDB, Cassandra e Redis. Para escolher o melhor banco de dados, seja interativo ou não interativo, devemos definir a finalidade e o tipo de dados com os quais trabalhamos, pois isso determinará o uso dos dados. Portanto, se você estiver trabalhando, por exemplo, com Big Data e tipo de dados JSON, uma boa opção seria usar um banco de dados não relacional, pois ele foi criado para lidar com esse tipo de dados.