INTRODUÇÃO

Um banco de dados é uma maneira de armazenar informações para que os dados possam ser armazenados e recuperados sempre que necessário. Um sistema de gerenciamento de banco de dados relacional é um típico banco de dados cujas informações são armazenadas em tabelas. A maioria dos bancos de dados utilizados como parte das organizações hoje em dia são bancos de dados relacionais, em vez de um sistema de arquivos plano ou quaisquer outros bancos de dados. Os bancos de dados relacionais têm a capacidade de lidar com um grande número de informações e consultas complexas. As informações são guardadas em partes e montes de tabelas, ou 'relações'. Essas tabelas são isoladas em linhas e segmentos. No entanto, com a explosão do volume de dados, a consulta de informações baseadas em SQL tornou-se um desafio significativo no manuseio de bancos de dados maiores.

Um banco de dados NoSQL (originalmente chamado de "não-SQL" ou "não-relacional") fornece um mecanismo para armazenar e recuperar dados do modelo de maneiras diferentes das relações de tabela usadas em bancos de dados relacionais. O principal objetivo do movimento NoSQL é facilitar o armazenamento e a recuperação de dados, independentemente de sua estrutura e conteúdo. Eles são considerados alternativas para superar as restrições do atual cenário de persistência dominado pelo SQL e, portanto, também são conhecidos como bancos de dados não relacionais. As estruturas NoSQL são bancos de dados não relacionais transmitidos, destinados ao armazenamento de informações em escala expansiva e ao manuseio de informações muito paralelas em inúmeros servidores. Eles também utilizam dialetos e instrumentos não-SQL para se comunicar com informações.

Mesmo que esses dois tipos (SQL, NoSQL) difiram em muitos aspectos, dependendo da implementação, eles podem ser usados para aplicativos semelhantes, embora não seja recomendado, pois um não se destina como uma alternativa ao outro.

É evidente que, no contexto atual, há um crescimento exponencial do volume de dados estruturados e não estruturados (Big Data) de uma variedade de fontes de dados, como mídias sociais, e-mails, documentos de texto, dados GPS, dados de sensores, dados de vigilância, etc. Podemos, portanto, dizer que o Big Data é caracterizado por dados estruturados, semiestruturados e não estruturados de recursos digitais e não digitais. O principal desafio é usar efetivamente o Big Data, que é a fonte de dados para uma tomada de decisão eficiente, usando tecnologia apropriada para mineração de dados. As dificuldades atuais do Big Data se devem às seguintes características gerais dos negócios :

- ***Alta velocidade de dados -*** fluxos de informações de diferentes fontes e locais atualizados de forma rápida e contínua.
- ***Variedade de Dados -*** armazenamento de dados estruturados, semiestruturados e não estruturados.

- ***Volume de dados -*** um grande número de conjuntos de dados de vários tamanhos de terabyte ou petabyte.
- ***Informações de complexidade -*** dados organizados em vários locais ou centros de informação. Análise de Big Data, o processo pelo qual conjuntos de Big Data que compreendem uma variedade de tipos de dados são examinados pelas empresas.

Usando Big Data Análises, as empresas podem analisar enormes volumes de dados com mais precisão para revelar padrões ocultos, correlações desconhecidas, tendências de mercado, preferências para os consumidores e outras informações comerciais úteis. A análise de Big Data depende de grandes volumes de dados que exigem clusters para o estoque de dados para apoiar a tomada de decisões oportuna e eficaz. Como os bancos de dados relacionais não se aplicam a clusters e problemas de eficiência de exibição relacionados à análise de dados grandes, as empresas consideram a necessidade do movimento NoSQL.

O esquema NoSQL não está fixo. Ele utiliza diferentes interfaces para armazenar e analisar a quantidade pura de material gerado pelo usuário, informações privadas e informações espaciais geradas por aplicativos avançados, computação em nuvem e dispositivos inteligentes.

Neste contexto, o NoSQL DB fornece uma solução preferida do que o SQL DB, principalmente por causa de sua capacidade de lidar com particionamento horizontal de informações, processamento dinâmico de dados e melhoria de desempenho. Grandes empresas de internet (Facebook, LinkedIn, Amazon e Google) que não conseguem gerenciar serviços através do uso de bancos de dados relacionais atuais estudaram e levaram o NoSQL a corrigir seu problema com gerenciamento de informações em constante crescimento, uso otimizado de informações e escalabilidade horizontal de informações em larga escala. Para sistemas de TI com alta eficiência e dinâmica, o banco de dados NoSQL é a melhor escolha em comparação com a confiabilidade e um caráter extremamente distribuído de sistemas de arquitetura de Internet de três camadas e computação em nuvem.

BANCOS DE DADOS RELACIONAL

Os dados foram originalmente armazenados em documentos. No entanto, à medida que a quantidade de informações aumentou, acessar as informações usando arquivos não foi fácil. Era um método lento e ineficiente. Bancos de dados hierárquicos e de rede foram concebidos como mecanismos de armazenamento, mas não forneceram uma técnica normal para acesso a dados.

O SQL surgiu com a necessidade de lidar com informações e o desejo de uma técnica normal de acesso a informações.

Quando um sistema de transação faz qualquer transação, o sistema tem que garantir que a transação atenda a certas características. A seguir estão algumas propriedades que devem ser cumpridas quando uma transação é feita:

- ***Atomicidade:*** Toda transação é atômica, ou seja, se uma parte do sistema falhar, todo o sistema falha.
- ***Consistência:*** Cada transação está sujeita a um conjunto de regras.
- ***Isolamento:*** Nenhuma transação interfere em outra transação.
- ***Durabilidade:*** Se alguma pessoa estiver comprometida com a transação, outra pessoa receberá os mesmos dados comprometidos.

O ACID é essencial, mas somente quando o sistema é um tipo de sistema bancário, financeiro, de segurança, etc., que pode ser sobrecarga para aplicativos que precisam compartilhar grandes quantidades de informações, como Google, Amazon etc. Para alguns dos seguintes requisitos, não se encaixa bem:

1. Distribuído
2. Escalabilidade
3. Controle sobre as características de desempenho
4. Alta disponibilidade
5. Baixa Latência
6. Cheap

BANCOS DE DADOS NOSQL

À medida que um ambiente tecnológico se transforma e enfrenta novas dificuldades, as empresas reconhecem progressivamente que novos métodos e bancos de dados precisam ser avaliados para lidar com suas informações para ajudar a mudar as necessidades da empresa e aumentar a complexidade e o desenvolvimento.

O Banco de Dados Relacional foi o modelo dominante para administração de banco de dados. Mas os bancos de dados não relacionais, em nuvem ou "NoSQL" agora estão emergindo em comum como um modelo alternativo para o gerenciamento de bancos de dados.

O principal motivo por trás dessa estratégia é: design mais simples, escala "horizontal" mais simples para Clusters de máquinas, que é um problema para bancos de dados relacionais e melhor controle de acessibilidade. As estruturas de informação usadas em bancos de dados NoSQL (por exemplo, valor-chave, gráfico ou documento) são ligeiramente diferentes daquelas usadas em bancos de dados relacionais por padrão, tornando algumas atividades no NoSQL mais rápidas. As estruturas de informação usadas em bancos de dados NoSQL às vezes também são consideradas "mais flexíveis" do que em tabelas de banco de dados relacionais. No entanto, suas capacidades totais ainda não são divulgadas.

Em Big Data e aplicativos da web em tempo real, os bancos de dados NoSQL estão sendo cada vez mais usados. Para enfatizar que eles podem suportar SQL - como linguagens de consulta, os sistemas NoSQL às vezes também são chamados de "Não apenas SQL".

Muitos bancos de dados NoSQL são acessíveis, mas se enquadram em quatro modelos de dados descritos abaixo. Cada categoria tem suas próprias características particulares, mas os modelos de informação distintos são verificados. Todos os bancos de dados NoSQL são geralmente projetados para distribuição e dimensionamento horizontal, não expõem uma interface SQL e podem ser de código aberto. Os bancos de dados NoSQL variam dependendo de seu modelo de dados em seu desempenho.

Banco de dados de armazenamento de documentos

O banco de dados de armazenamentos de documentos refere-se a bancos de dados nos quais as informações são armazenadas na forma de Documentos. As lojas de documentos oferecem excelente eficiência e opções para escalabilidade horizontal. Os documentos dentro de um banco de dados orientado a documentos são comparáveis aos documentos em bancos de dados relacionais, mas são muito mais flexíveis porque são menos esquemáticos. Os formatos padrão dos documentos são como XML, PDF, JSON e assim por diante.

Em bancos de dados relacionais, um registro dentro do mesmo banco de dados terá os mesmos campos de dados e os campos de dados não utilizados serão mantidos vazios, mas cada documento pode ter dados semelhantes e diferentes no caso de armazenamentos de documentos. Uma chave exclusiva que representa o documento é usada para abordar documentos no banco de dados. Essas chaves podem ser uma string simples ou um URI ou string de caminho. Em comparação com os armazenamentos de valores-chave, o armazenamento de documentos é um pouco mais complexo porque eles permitem que os pares de valores-chave sejam incorporados em documentos que também são conhecidos como pares de documentos-chave.

Para sistemas de gerenciamento de conteúdo e aplicativos de blog, bancos de dados orientados a documentos são adequados. Exemplos são: MongoDB, Apache CouchDB, DocumentDB do Azure e AWS DynamoDB, provedores que usam bancos de dados orientados a documentos. O MongoDB é desenvolvido com um C++ de 10 G e é um banco de dados baseado em placas e orientado a documentos entre plataformas. O Grid File System é usado para armazenar arquivos grandes em formato JSON binário, como imagens e vídeos. Ele oferece alta eficiência, consistência e persistência, mas não é muito confiável.

Banco de dados de armazenamentos de valores-chave

Os armazenamentos de dados com valor-chave são muito simples, mas são silenciosamente eficazes e fortes. A interface do programa de aplicativos (API) é fácil de usar. O usuário pode salvar os dados de forma menos esquematiza usando o armazenamento de dados de valor-chave. Os dados são geralmente um tipo de linguagem de programação ou tipo de objeto de dados. A informação consiste em 2 componentes, uma string que descreve a chave e o valor real, produzindo um par de "valor principal". Os salvamentos de dados são como tabelas de hash nas quais as chaves são usadas para indexação, tornando-as mais rápidas do que o banco relacional.

O modelo de dados é, portanto, simples: um mapa e um dicionário que permite ao usuário solicitar valores com base em valores-chave especificados. Nos armazenamentos de dados modernos, a escalabilidade das informações é preferível à consistência. Portanto, consultas ad-hoc e características analíticas, como links e agregados, foram negligenciadas. As lojas de valor-chave oferecem alta competitividade, pesquisa rápida e opções de armazenamento em massa. Uma das fraquezas do armazenamento de dados chave é a ausência de um esquema para criar uma visão personalizada dos dados.

Esses bancos de dados de valor-chave podem ser usados como carrinhos de compras on-line para criar fóruns e sites para armazenar sessões de clientes. DynamoDB, Cassandra, Azure Table Storage (ATS) da Amazon são alguns exemplos notáveis. Para aplicativos em escala de internet, a Amazon fornece o Serviço de Loja NoSQL totalmente controlado do DynamoDB. É uma instalação de armazenamento de valor-chave distribuído que, com sua função de réplica, oferece acesso rápido, seguro e econômico a informações e alta disponibilidade e durabilidade.

Banco de Dados Gráfico

Banco de dados de gráficos são bancos de dados que armazenam informações como gráficos. O gráfico contém nós e bordas, que mantêm as relações entre os nós e os itens. O gráfico também inclui características relacionadas a nós. Ele utiliza um método de adjacência livre de índice, o que significa que cada nó compreende um ponto direto que aponta para o nó vizinho.

Este método permite que milhões de documentos sejam acessados. O foco principal na associação entre informações, em um banco de dados gráfico. Os bancos de dados gráficos fornecem armazenamento de dados esquemáticos e semiestruturados menos eficaz. As consultas são articuladas como crossover, aumentando assim a velocidade dos bancos de dados gráficos sobre os bancos de dados de relações. É simples de medir e simples de usar. Os bancos de dados gráficos estão em conformidade com o ACID e promovem a reversão.

Essas bases de dados são projetadas para o desenvolvimento de aplicativos de redes sociais, bioinformática, sistemas de gerenciamento de conteúdo e serviços

de gerenciamento em nuvem. Bancos de dados de gráficos notáveis são Neo4j, Orient DB, Apache Giraph e Titan.

Banco de dados de lojas de coluna ampla

Os armazenamentos de colunas NoSQL são armazenamentos de linhas / colunas híbridas em oposição a bases relacionais puras. Embora o armazenamento de dados coluna por coluna e as adições de coluna a bancos de dados baseados em linhas sejam compartilhados, as lojas de colunas não armazenam informações de banco de dados em listas, mas armazenam as informações em arquiteturas distribuídas massivamente. Cada chave tem uma ou mais características (linhas) para cada linha em lojas de colunas. Uma loja de colunas armazena suas informações para que menos atividade de E/S possa ser adicionada rapidamente. Ele fornece forte escalabilidade de armazenamento de dados. As informações salvas no banco de dados são baseadas na ordem de classificação da família de colunas.

Bancos de dados de coluna ampla são perfeitos para mineração de dados e aplicativos de análise de Big Data. Exemplos de fornecedores de lojas orientadas a colunas incluem Cassandra (o alto desempenho do Facebook), Apache Hbase, Big Table do Google e HyperTable. O Big Table do Google é um banco de dados de alto desempenho de ampla coluna, capaz de lidar com grandes quantidades de informações. Foi criado usando C / C++ no Google File System GFS. É usado por vários aplicativos do Google, como YouTube e Gmail, que têm diferentes requisitos de latência do banco de dados. Além do uso no Google App Engine, ele não é distribuído fora do Google. O Big Table foi concebido para escalabilidade simples em milhares de computadores, por isso é tolerante ao hardware.

Características dos Bancos de Dados NoSQL

Teorema CAP - Em essência, os bancos de dados NoSQL selecionam dois dos três princípios do teorema CAP (consistência, disponibilidade, tolerância à partição). Para obter melhor disponibilidade e particionamento, muitos bancos de dados NoSQL afrouxaram as demandas de consistência. Isso resultou em sistemas BASE (basicamente disponíveis, estado suave, em última análise consistente). Isso implica que um compromisso pode ser feito, por exemplo, com baixo desempenho ou oferecer alta acessibilidade e baixa consistência com desempenho rápido.

Escalável vertical e horizontalmente - Tradicionalmente, os bancos de dados relacionais vivem em um único servidor que pode ser dimensionado para fornecer escalabilidade, adicionando mais processadores, armazenamento e memória. A replicação é geralmente usada para manter os bancos de dados sincronizados em bancos de dados de relacionamento residentes em vários servidores. Os bancos de

dados NoSQL podem estar em um único servidor, mas são projetados com mais frequência para operar em uma nuvem de servidores.

Armazenamento em disco – Este conjunto de dados vivem quase sempre em uma unidade de disco ou rede de zona de armazenamento. Como parte do SQL, selecione ou salve as atividades de procedimento, os conjuntos de linhas do banco de dados são colocados na memória. Alguns bancos de dados NoSQL são construídos para existir na memória para velocidade e podem continuar a ser armazenados no disco.

Os recursos do NoSQL que pesquisamos são:

- Persistência
- Replicação
- Alta Disponibilidade
- Transações
- Conscientização de localização em rack
- Econômico
- Aplicativos de Big Data
- Escalável

COMPARAÇÃO ENTRE BANCO DE DADOS RELACIONAIS E BANCOS DE DADOS NOSQL

A principal razão pela qual alguém precisaria mudar para bancos de dados NoSQL é a necessidade de armazenamento de dados enormes (também chamado de Big Data), escalabilidade e por razões de desempenho. Aqui estão algumas tabelas que exibem a diferença na terminologia e nas operações de dados entre um banco de dados NoSQL e SQL.

SQL	MONGODB
Database	Database
Table	Collection
Row	document or BSON document
Column	Field
Index	Index
table joins	embedded documents and linking
primary key (specify any unique column or column combinations as primary key)	primary key (the primary key is automatically set to the _id field in MongoDB)
aggregation (e.g. by group)	aggregation pipeline

Table 2. Select query of SQL and MongoDB

SQL	MongoDB
Select * from employees	db.employees.find()

Table 3. Insert query of SQL and MongoDB

SQL	MongoDB
INSERT INTO employees VAL UES("wajid","335","BS")	db.employees.insert(name : "wa jid", Roll: "335", degree: "BS")

Table 4. Create query of SQL and MongoDB

SQL	MongoDB
Create Table employees(id int,name varchar(20),roll int)	db.createCollection("employees")

Table 5. Drop query of SQL and MongoDB

SQL	MongoDB
DROP TABLE employees	db.employees.drop()

Table 6. Summarized view of SQL and NoSQL

SQL	NoSQL
Based on ACID transactional properties such as atomicity, consistency, isolation.	Supports AID transactions and CAP theorem of distributed systems support consistency of data across all nodes of a NoSQL database.
It has vertical Scaling.	It has horizontal Scaling.
Structured Query Language are used to manipulate the data.	Query the Data efficiently. Object oriented APIs are used.
Based on pre-defined foreign keys relationships between tables in an explicit database schema. Strict definition of schemas and data type is required before inserting the data.	Dynamic database schema. Do not force schema definition in advance. Different data can be store together as required.
Softwares that use for this DB are oracle, MySQL, SQL Server.	MongoDB, Riak, Couchbase, Cassandra.

DESEMPENHO DE BANCOS DE DADOS NOSQL E SQL PARA ANÁLISE DE BIG DATA

Benefícios NoSQL sobre SQL

- Fornece uma ampla seleção de modelos de dados

- Facilmente escalável
- Os administradores do banco de dados não são necessários
- Alguns fornecedores NoSQL DB, como Riak e Cassandra podem gerenciar falha de hardware
- Mais rápido, mais eficiente e flexível
- Desenvolveu-se muito rapidamente
- Usado para aplicativos de big data

Desvantagens NoSQL sobre SQL

- Imaturidade
- Sem idioma de consulta padrão
- A compatibilidade ACID não é possível em alguns bancos de dados NoSQL
- Sem especificação de interface.
- É difícil de manter
- Menor suporte

Benefícios do SQL sobre o NoSQL

- É simples de usar
- Fácil de projetar, executar, manter e usar
- Apenas as informações são salvas em um lugar
- Ele oferece uma variedade de interfaces
- Melhora a integridade das informações
- É protegido por natureza
- Tem linguagem de consulta padrão

Desvantagens do SQL sobre o NoSQL

- O software é caro
- Alta despesa em hardware
- Limitações na estrutura
- Os dados perdidos dificilmente podem ser recuperados
- Problema de alta disponibilidade
- Não suporta aplicativos de Big Data
- Processamento lento em certas aplicações

DATA LAKE E DATABRICKS

Data Lake é um conceito relativamente novo, sendo definido como “uma metodologia habilitada por um enorme repositório de dados baseado em tecnologias de baixo custo que melhora a captura, refinamento, arquivamento e

exploração de dados brutos dentro de uma empresa". Um Data Lake pode conter dados brutos, não estruturados ou multiestruturados, onde a maior parte desses dados pode ter valor não reconhecido para a organização.

A ideia básica do Data Lake é simples: todos os dados emitidos pela organização serão armazenados em uma única estrutura de dados chamada Data Lake. Os dados serão armazenados no lago em seu formato original. O pré-processamento complexo e a transformação de dados de carregamento em data warehouses serão eliminadas. Os custos iniciais da ingestão de dados também podem ser reduzidos. Uma vez que os dados são colocados no lago, eles estão disponíveis para análise por todos na organização:

- Todos os dados são carregados a partir de sistemas de origem.
- Nenhum dado é recusado.
- Os dados são armazenados no nível da folha em um estado não transformado ou quase não transformado.

Quando se trata do núcleo, há apenas duas operações no processamento de dados, Transacional e Analítico. As operações diárias, como o Processamento Transacional On-line (OLTP), funcionam principalmente com as operações CRUD-Create, Replicate, Update, Delete dos dados para rotinas diárias.

Atualmente, os data warehouses são a abordagem dominante para o fornecimento de dados analíticos. Apenas os dados transformados serão armazenados no data warehouse. O data warehouse é criado com base na tabela de fatos com perguntas simples: "quem, o que, quando, onde". Em seguida, as tabelas de dimensão são complementadas com base nos campos dos bancos de dados. Os dados são extraídos, transformados para estar em conformidade com o esquema de data warehouse e carregados no Data Warehouse (operações ETL).

A empresa reúne dados de vários bancos de dados operacionais em um único armazenamento de data warehouse para executar consultas ad-hoc. A consulta executada nos dados consolidados pode ajudar a recuperar a inteligência de negócios de forma conveniente. Isso causa a separação de preocupações para os dois conceitos importantes. As transações diárias serão realizadas por sistemas como o OLTP (processamento de Transações Online). O processo analítico, como análise de dados, revisão de dados históricos e dados correlacionados, será realizado por um sistema analítico como o OLAP. Os dados das transações permanecem em bancos de dados operacionais, enquanto consultas ad-hoc complexas serão executadas em armazéns de dados que servem para fins analíticos (assim, não degradarão o desempenho, por exemplo, tempo de resposta da consulta de transações). Como o DW se destina a construir para fins analíticos, os dados são carregados no modo de lote com os intervalos regulares definidos. O processo de análise de dados é realizado nos dados armazenados no data warehouse para apoiar a tomada de decisões da empresa e obter a valiosa visão de negócios.

No entanto, como os data warehouses são muito grandes e levam tempo para serem criados, “Data Marts” podem ser criados. Os “Data Marts” são menores que os data warehouses e destinam-se a armazenar os dados de uma parte da organização (ou seja, um departamento na empresa). O data warehouse armazenará os dados de toda a empresa. Esses mercados de dados podem ser construídos separadamente. Ou uma parte do data warehouse destinada a funcionalidade ou departamento específico pode ser extraída para criar um mercado de dados.

Os armazéns de dados precisam usar operações como agregado, junção, etc. em uma palavra - precisam calcular a carga intensiva de consultas. Portanto, a maioria dos armazéns de dados é construída como uma única estrutura de armazenamento consolidada, altamente resumida, dados ETL de vários bancos de dados transacionais.

Data lake (implementação arquitetônica)

Cada entidade de dados no lago está associada a um identificador exclusivo e um conjunto de metadados estendidos, e os consumidores podem usar esquemas específicos para consultar dados relevantes, o que resultará em um conjunto menor de dados que pode ser analisado para ajudar a responder à pergunta de um consumidor.

Muitas implementações do Data Lake são originalmente baseadas no Apache Hadoop. Uma variedade de dados de armazenamentos de dados heterogêneas será extraída para ser armazenada no Hadoop Cluster. HADOOP (Highly Available Object Oriented Data Platform) é uma ferramenta de big data amplamente popular, especialmente adequada para a carga de trabalho de processamento em lote de big data. O Hadoop tem dois componentes principais - HDFS (Hadoop Distributed File System) e o mecanismo MapReduce. O Sistema de Arquivos HDFS lida com o único ponto de falha e escalabilidade replicando várias cópias de blocos de dados em diferentes nós do cluster. Todos os dados armazenados nesses blocos de dados serão processados na abordagem MapReduce. Os dados serão recuperados como uma lista de pares chave-valor, ou seja, Fase do mapa. As mesmas chaves de dados serão embaralhadas, classificadas e listadas em grupos para executar as operações necessárias, ou seja, reduza a fase. Todos os dados produzidos por uma empresa serão despejados no Data Lake Hadoop Cluster.

Para o carregamento em tempo real, os lagos de dados posteriores estão usando a estrutura de processamento de fluxo, como Apache Spark, Apache Flink. Os dados necessários serão transformados de acordo com a necessidade dos sistemas de análise em tempo real no tempo de consulta. Salvar dados com vários formatos e estruturas de fontes heterogêneas e lidar com diferentes velocidades de dados (ou seja, diferentes velocidades de processamento de big data) exige a cuidadosa consideração na construção de tubos de dados para transportar dados para o lago.

Pode-se dizer que a estratégia do Data Lake inclui o armazenamento de todos os tipos de dados (variedade de dados) de bancos de dados SQL e NoSQL, bem como a combinação dos conceitos de OLTP com OLAP. Os bancos de dados SQL são usados para armazenar dados estruturados. Os bancos de dados NoSQL (Key-value, Columnar, Document e Graph Stores) são usados para armazenar dados semiestruturados e não estruturados. No entanto, eles também podem ser usados para armazenar dados estruturados. Todos os dados transacionais desses bancos de dados (Extrair - E) serão armazenados (Carregar - L) no data lake sem alterar seu formato. Quando os dados são necessários (tempo de consulta), os dados no lago serão transformados (Transformar - T) de acordo com as partes do sistema corporativo. Os trabalhos necessários para operações de consulta precisam ser executados no nível do aplicativo.

As chaves para criar um lago de dados bem-sucedido são sugeridas da seguinte forma:

1. Alinhe a iniciativa de inovação com a estratégia corporativa. A prioridade da estratégia corporativa é Aceleração de Negócios, Eficiência Operacional, Segurança e Risco. A implementação do Data Lake deve se concentrar na principal prioridade da estratégia corporativa.
2. Aplique uma estratégia sólida de integração de dados. A tecnologia para integração de dados pode estar mudando as horas extras no big data. As primeiras soluções Data Lake são baseadas no Hadoop. Para lidar com dados em tempo real e streaming, muitas soluções DL agora estão usando a estrutura de streaming, como Spark, Flink. Os Data Lakes precisam acompanhar as melhores práticas em evolução para o gerenciamento de metadados. O pipeline analítico de dados precisa automatizar o processo de extração, carregamento, limpeza, transformação e análise de dados.
3. Estabeleça uma estratégia de integração moderna. O Data Lake pode ser preenchido em carga em lote ou em alimentação de gotejamento. Simplifique o processo de carregamento de dados (independentemente de tipos, fontes ou complexidade) no lago, permitindo e estabelecendo processos repetíveis. Enquanto isso, mantenha o nível apropriado de governança de dados. Preste atenção ao processo de injeção de metadados em tempo real.
4. Abrace novas estratégias de gerenciamento de dados adotando a ingestão precoce e o processamento de execução adaptativo, como MapReduce, Spark ou Flink, que permitem flexibilidade. Derivar metadados no tempo de integração (carregamento). Crie o modelo analítico em tempo real (automatize o processo de criação). Estenda o processamento e as estratégias de gerenciamento de dados a todos os dados. A análise de dados deve ser capaz de ser aplicada em qualquer lugar do pipeline de dados. Modernize a infraestrutura de integração de dados.
5. Aplique algoritmos de aprendizado de máquina para gerar valor comercial real. O fluxo de trabalho para aplicar algoritmos de aprendizado de máquina deve ser repetível. O fluxo de dados deve acompanhar a preparação de dados, o recurso de dados de engenharia e a manipulação de conjuntos de dados.

Os conceitos do Data Lake obviamente se desviam do data warehouse por meio do processamento de dados na ordem “E-T-L” e utilizando a abordagem “Schema-on-Read”. As abordagens de data warehouse seguem o processo tradicional de ETL. Primeiro, os dados de bancos de dados operacionais são extraídos (E). Em seguida, os dados são processados, limpos e transformados (T) antes de carregá-los (L) nos data warehouses ou data marts. O armazenamento de dados é especialmente projetado para lidar com carga de trabalho pesada de leitura para análise. O data warehouse precisa definir seu esquema com antecedência antes que os dados sejam carregados. Portanto, eles são considerados como uma abordagem “Schema-On-Write”.

Diferente da ideia tradicional de ETL de data warehouses, os Data Lakes fazem os diferentes pedidos no processamento de dados. Os dados serão armazenados em seu formato original. A etapa de pré-processamento não será tratada até que os dados sejam exigidos pelo aplicativo ou no tempo de consulta. Portanto, ele quebra as regras tradicionais de ETL do data warehouse. Em vez disso, o DL promove a nova ideia de ELT (Extrair, Carregar, Transformar) a mudança de ordem para processamento de dados (ELT). Não há nenhum esquema de dados predefinido no DL. Quando os dados são extraídos da fonte para o data lake, os metadados necessários são especialmente adicionados aos dados. Dessa forma, o Data Lake pode lidar com carga de trabalho pesada de gravação e carga de trabalho pesada de leitura (transação e análise), recombinando duas separações de preocupações (escrita e leitura). Os dados não são transformados até que os aplicativos os chamem. Somente quando os dados são necessários e chamados pelo aplicativo para consulta, os dados são transformados em forma apropriada usando os metadados adicionados anteriormente. Dessa forma, a pré-transformação de dados caros pode ser evitada no Data Lake. As operações de transformação só serão realizadas quando os dados forem lidos do data lake. Portanto, a abordagem do Data Lake é chamada de abordagem “Schema-on-Read”.

Comparison	Data Warehouse	Data Lake
Data	Structured, processed data	Structured/semi-structured, unstructured data, raw data, unprocessed data
Processing	Schema-on-write	Schema-on-read
Storage	Expensive, reliable	Low cost storage
Agility	Less agile, fixed configuration	High agility, flexible configuration
Security	Matured	Maturing
Users	Business professional	Data Scientists (especially those familiar with domain)

Dados - Uma palavra popular no domínio empresarial é que apenas 20% dos dados estão estruturados. O DW é a nata da cultura, pois só aceitava dados estruturados

ETL-ed e consolidados extremamente resumidos. Os outros 80% dos dados semiestruturados e não estruturados (não há sugestão exata de quantos % cada um deles ocorre - apenas a maior parte não é estruturada e a parte semiestruturada é maior que os dados estruturados. Outros especialistas preveem como 70% dos dados não estruturados + semiestruturados.) Como o DL armazena dados de maneira não estruturada e organizada, eles podem ser mais facilmente manipulados e tratados de várias maneiras, o que é surpreendentemente mais adequado para big data.

Processamento - Como mencionado anteriormente, os dados armazenados no data warehouse são cuidadosamente selecionados através do processo Extract-Transform-Load. Eles destinam-se a ser usados apenas para fins analíticos - especialmente para o nível do tomador de decisão. Eles passaram pela ordem Extract-Transform-Load de processamento de dados com a abordagem Schema-on-Write. No entanto, os data warehouses são limitados por sua natureza muito resumida e estruturada. Eles não são capazes de responder a perguntas fora da caixa de tomadores de decisão ou perguntas que precisam extrair dados de transações e/ou combinadas com dados não estruturados. A consulta do usuário só pode ser realizada de forma limitada em dados estruturados altamente definidos. No entanto, o Data Lake pode lidar com esse tipo de consulta do usuário. Somente quando o usuário consultar os dados, os dados serão transformados de acordo com os aplicativos do usuário para o nível de análise na ordem de processamento (Extract-Load-Transform) aplicando a abordagem "Schema-on-read". Isso dá mais flexibilidade para os cientistas de dados que estão familiarizados com o domínio para recuperar valor de dados anteriormente inexplorados ou inexplorados, como dados brutos ou binários, combinados com dados estruturados. Também é uma razão pela qual os cientistas de dados são capazes de tolerar as imperfeições do Data Lake e querem adotar e refinar o Data Lake.

Custo - Muitas soluções de data lake são implementadas em uma estrutura de código aberto e projetadas para servidores de commodities. Portanto, em comparação com as altas taxas de licença de armazenamento de data warehouse, é relativamente muito mais barato. De acordo com o custo do data warehouse pode ser recuperado dentro de um ano. No entanto, o custo do Data Lake e seu ROI ainda é um debate questionável.

Agilidade - O design do data warehouse é feito com antecedência antes que os dados sejam carregados (esquema na gravação). Por definição, é uma definição altamente estruturada com gerenciamento de dados altamente governado. Embora seja possível alterar o design do data warehouse, é muito demorado e requer um enorme esforço, pois está vinculado a muitos processos de negócios. Às vezes, toda a reconsideração do projeto é necessária, levando a um custo de manutenção significativo. Além disso, o data warehouse não pode lidar com a solicitação de empresas que desejam analisar uma ampla variedade de dados. Se eles forem forçados, os dados já carregados no data warehouse podem estar corrompidos ou o design do Armazém pode estar danificado. O Data Lake não tem a estrutura explicitamente definida do data warehouse. Portanto, eles são mais flexíveis e

ágeis. Ele dá aos desenvolvedores e cientistas de dados a capacidade de configurar facilmente os modelos, consultas e aplicativos em tempo real.

Segurança - Os DW de dados estão lá há décadas e têm segurança bem definida. No entanto, os DL ainda não têm a resposta para a pergunta completa de segurança “quando”. A segurança dos dados no Data Lake ainda é deixada como área de pesquisa aberta.

Usuários - Até agora, o Data Lake ainda é o mais adequado para analistas de dados e cientistas de dados, não análises para todos devido às razões mencionadas acima.