Kaique H. Valim RM 346269 MBA em Data Science - 2DTSR

Data Architecture I **Trabalho Final**

1. **O que são dados transacionais?**

São informações capturadas das transações. Por exemplo, estes dados registram a hora da transação, o local onde ocorreu, os preços dos itens comprados, a forma de pagamento empregada, os descontos aplicados e demais informações necessárias para as áreas do negócio. Portanto dados transacionais são os dados gerados pelos aplicativos durante a execução ou suporte a processos comerciais diários de uma empresa. Esses dados transacionais costumam ser categorizados como dados estruturados.

2. **Defina um RDBMS e cite alguns exemplos de softwares existentes de RDBMS.**

RDBMS organizam os dados como uma série de tabelas bidimensionais como linhas e colunas. Onde cada linha em uma tabela é um registro e deve garantir uma identidade única. Um RDBMS implementa um mecanismo consistente de forma transacional que deve estar em conformidade com o modelo ACID (Atômico, Consistente, Isolado e Durável).

Um RDBMS precisa fornecer suporte para um modelo de esquema em gravação, onde a estrutura de dados é definida antecipadamente e todas as operações de leitura ou gravação devem utilizar o esquema.

Softwares: Microsoft SQL Server, Google Cloud Platform, Oracle Database, MariaDB, PostgreSQL.

3. **O que significa cada componente ACID?**

A *atomicidade* define todos os elementos que compõem uma transação completa do banco de dados. A *consistência* define as regras para manter os pontos de dados em um estado correto após uma transação. O *isolamento* mantém o efeito de uma transação invisível para outras pessoas até ser confirmada, para evitar confusão. A *durabilidade* garante que as alterações de dados se tornem permanentes quando a transação for confirmada.

4. **O que é normalização?**

A normalização é um processo a partir do qual se aplicam regras a todas as tabelas do banco de dados com o objetivo de evitar falhas no projeto, como redundância de dados e mistura de diferentes assuntos numa mesma tabela.

5. **Conceitue Primary Key, Unique Key, Foreign Key e Not Null.**

PRIMARY KEY é uma regra que declara que uma ou mais colunas serão utilizadas para identificar unicamente cada registro (linha) desta tabela e, também, nenhum valor nulo será aceito neste caso. Sempre um índice será criado para garantir a regra da chave primária, agilizando a verificação da unicidade dos dados da chave. *UNIQUE* é uma regra que determina que não será permitido valor duplicado, mas aceita valor nulo. O Índice é criado automaticamente. *FOREIGN KEY* permite você referenciar uma chave primária da mesma ou de outra tabela, regra utilizada para criação do relacionamento entre tabelas. *NOT NULL* é uma regra que declara que a coluna não aceita valor nulo, ou seja, fica sem um valor atribuído.

6. **O que é Particionamento no MySQL e quais os tipos existentes?**

Particionamento no MySQL permite a distribuição dos dados de uma tabela por vários arquivos a partir de regras definidas. A utilização do particionamento visa a distribuição de grandes volumes de dados e de sistemas que exigem desempenho por parte dos sistemas de gerenciamento de banco de dados em busca por frações destes dados. O particionamento no MySQL aplica-se para todos os dados e índices, não é possível particionar apenas os dados e não particionar os índices, ou vice-versa, nem mesmo particionar uma parte da tabela.

7. **O que é Data Warehouse e quais suas vantagens?**

Um Data Warehouse é um tipo de sistema de gerenciamento de dados, é um repositório centralizado que contém dados estruturados e dados semi estruturados para fins de relatórios, análise e atendimento das necessidades de inteligência do negócio (BI). Os dados fluem de uma variedade de fontes, como sistemas de vendas, compras, clientes, financeiro, RH, ou seja, dos seus bancos de dados relacionais ou arquivos que geralmente são limpos, padronizados e/ou customizados antes de serem carregados no DW. Desta forma, temos uma enorme quantidade de dados de histórico armazenados que serão minerados, consultados, analisados e visualizados pelas áreas de negócio. Alguns recursos precisam ser levados em consideração quando implantamos um DW: paralelismo, particionamento, compressão, índices especiais e ferramentas de visualização.

8. **Quais são os componentes básicos de um Data Warehouse?**

Um Data Warehouse costuma ser criado com os seguintes componentes: Um banco de dados relacional (RDBMS) para armazenar e gerenciar os dados; Uma solução de extração, carregamento e transformação (ETL) para preparar os dados para análise; Análise estatística, relatórios e recursos de mineração de dados; Ferramentas de análise de clientes para visualizar e apresentar dados aos usuários de negócios.

9. **Conceitue e diferencie tabelas Fato e Dimensão.**

As medidas representam os dados de fato, por isso são chamados de “fatos”. O termo “fato” representa uma medida do negócio. Estes dados estão organizados por uma ou mais dimensões. Todas as tabelas fatos costumam ter duas ou mais FKs, que conectam às suas tabelas de dimensão. São as células que preenchem o cubo. Os dados podem ser numéricos, texto, dados, booleanos etc. Características das tabelas Fatos: Pode conter medidas numéricas de negócio; Contém um grande volume de dados; Pode conter dados básicos, derivados ou sumarizados; São unidas as tabelas de dimensões.

As dimensões dão um significado, um sentido ao fato. As dimensões costumam descrever: “quem, o quê, onde, quando, como e por quê” São categorizadas: Identificar e categorizar seus dados medidos; Eles moldam as medidas formando as bordas das medidas. Outra característica importante é que as dimensões podem ser compartilhadas com outras tabelas de fato. Elas são criadas e armazenadas uma vez mais, usadas repetidamente. Características das tabelas de dimensão: Pode conter uma ou mais hierarquias que categorizam os dados; Os atributos ajudam a descrever o valor dimensional.

10. **O que significa ETL e o que fazemos em cada etapa?**

ETL = Extract, Transform and Load. Extract: extrai dados da origem; Transform: processo de integração, verificação, validar, limpar, rotular (timestamp) e padronizar os dados. Load: carrega os dados da camada intermediária ou da fonte de dados para o DW.

11. Cite 5 considerações que devemos levar em conta ao planejar um DW.

Especificar os dados necessários; Reconhecer os relacionamentos críticos entre os grupos de dados; Definir o ambiente que irá suportar o Data Warehouse; Identificar a origem dos dados; Identificar as transformações necessárias; Especificar a frequência em que os dados precisam ser atualizados.

12. Quais são os tipos de operações que podemos executar em um Cubo.

Slice; Dice; Drill Down/UP; Roll-UP; Pivot

13. Conceitue Big Data.

Arquitetura sistêmica capaz de comportar dados com maior variedade, que chegam com volume crescente e com velocidade cada vez maior. Garantindo a veracidade e entregando valor.

14. Quais são os 5 Vs do Big Data?

Variedade; Volume; Velocidade; Veracidade; Valor .

15. O que é Data Lake?

Data Lake é o repositório central para todos os tipos de dados estruturados, semiestruturados e não estruturados. Os dados estão brutos, ou seja, armazenados da forma como estão. A complexidade de tratamento de dados aumenta devido a variedade. O Data Lake precisa ter governança, segurança, gestão dos metadados, monitoramento, eleição de Data Owners/Stewards e processos ETL complexos.

16. O que é Delta Lake?

Delta Lake é uma camada de armazenamento open-source que entrega confiabilidade, performance e segurança no seu Data Lake. Por uma fonte única, confiável e segura dos dados permite forçar esquemas e ACID transactions diretamente no lake

17. O que é Lakehouse?

Lakehouse é uma arquitetura aberta que combina os melhores elementos do Data Lake e Data Warehouse. Características: Suporte à transações; Governança;; Schema; Suporte à BI; Storage não está acoplado aos servidores; Suporte diversos tipos de dados; Suporta diversos tipos de workloads; Permite aplicações e relatórios real-time.

18. Quais são as categorias/divisões dos NoSQL?

Graph databases, Document databases, Key-value database e Column Family Store.

19. Por que devemos utilizar NoSQL?

NoSQL deve ser utilizado por apresentar as seguintes caracteristicas: Flexibilidade; Escalabilidade; Raízes Open-Source; Disponibilidade; Baixo custo Operacional; Funcionalidades Especiais; Schemaless; Não usa JOINS. Teorema de CAP.

20. Relacione as disciplinas de Governança de Dados.

Data Governance: oferece a direção e a supervisão para o gerenciamento de dados; *Data Architecture*: define um blueprint para gerenciar os ativos de dados; *Data Modeling & Design*: processo de descobrir, analisar, representar e comunicar os requerimentos de dados; *Data Storage & Operations:* define o ciclo de vida dos dados, como desenhar, implementar e suportar o armazenamento dos dados; *Data Security:* garante a privacidade, confidencialidade e como os dados devem ser acessados; *Data Integration and Interoperability:* processo relacionado ao movimento e consolidação dos dados entre locais de armazenamento, aplicações e organizações; *Document & Content Management:* planejamento, implementação e controle usado para gerenciar o ciclo de vida dos dados; *Reference & Master Data*: reconciliação e manutenção dos dados críticos para garantir a consistência, acuracidade, relevância dos dados; *DW & BI:* planejamento, implementação e controle para gerenciar os dados analíticos; *Metadata:* planejamento, implementação e controle de um catálogo de dados para entender mais sobre os dados e seu ciclo de vida; *Data Quality:* planejamento e implementação de técnicas para o gerenciamento da qualidade dos dados.