

## Enunciado

A QuantumFinance contratou a consultoria SmartMind Analytics para desenvolver seu aplicativo QuantumApp.

O objetivo é que o aplicativo seja simples e fácil de operar, e listaram uma série de requisitos para o negócio.

Como arquiteto de dados, desenvolva modelo de dados com a técnica mais performática para cada requisito do aplicativo:

1. Cadastro de cliente com fotos e assinatura eletrônica, e informações dos contatos do cliente
2. Processo de transações (Doc, TED, Pix)
3. Recomendação de produtos de crédito para a Home
4. Telas e relatórios para análise das transações da conta (exemplo: análise por período e por tipo de movimento)

**Entrega: diagramas de modelagem e scripts de criação dos bancos gerados automaticamente pelas ferramentas de modelagem.**

Para cada um dos cenários acima, descreva qual a modelagem mais indicada e o porquê. Logo em seguida, apresente um diagrama conforme a modelagem escolhida.

Se a sua modelagem for:

- Modelagem Relacional, então apresente um diagrama entidade-relacionamento;
- Modelagem Dimensional então apresente um diagrama apresentando as tabelas fato e dimensão;
- Modelagem NoSQL, do tipo Documento, então apresente um exemplo de JSON ou JSON Schema;
- Modelagem NoSQL, do tipo Grafo, então apresente um grafo contendo exemplos de instâncias como grafos e suas arestas;
- Modelagem NoSQL, do tipo Chave-Valor, então apresente uma lista contendo exemplos de chave-valor;
- Modelagem NoSQL, do tipo Colunar, então apresente exemplos de registros no estilo "família de super-colunas".

## Resposta:

Todos os quatro exemplos solicitados no enunciado possuem como característica serem informações que precisam ser consistentes e, muitas vezes, podem passar por processos de conciliação ou até mesmo serem auditáveis quanto à sua veracidade.

Visto isso, a escolha do grupo para os quatro problemas foi o uso de modelos relacionais, e explicaremos detalhadamente o porquê de cada uma das escolhas. Em anexo, encontra-se o script para criação de um banco de dados MySQL, juntamente com o modelo entidade-relacionamento e o modelo físico.

No primeiro caso, referente ao cadastro do cliente com fotos, assinatura e demais informações, compreendemos que alguns tipos de atributos não são suportados diretamente por bancos relacionais. No entanto, podemos salvar arquivos como imagens e

assinaturas em um bucket e guardar o caminho juntamente com a chave de descritografia, possibilitando o acesso posterior — sem a necessidade de realizar um workaround para utilizar esse tipo de banco. Acreditamos que o aspecto mais importante para esse tipo de informação é garantir que o cadastro do cliente, seja pessoa física ou jurídica, esteja correto e possa ser auditado quanto à veracidade dos dados. Utilizando o banco relacional, por garantir uma estrutura e padronização de dados, será possível manter as regras e normas estabelecidas pelo mercado para realizar o cadastro.

No segundo caso, na visão do grupo, o modelo ideal também é o relacional. Toda transação precisa ser rastreável e auditável para que seja possível realizar a conciliação futura. Os dados transacionais são a essência do projeto e, caso por algum motivo apresentem baixa confiabilidade, podem abrir margem para fraudes, sendo difícil identificar o problema posteriormente.

No terceiro caso, optamos novamente pelo modelo relacional, pois, no nosso entendimento, a visualização de um empréstimo em determinado espaço dentro do aplicativo depende da existência do empréstimo — não havendo possibilidade de essa relação não existir. Além disso, há outra relação em que o cliente precisa estar apto à contratação do empréstimo, sendo o setor de modelagem de crédito responsável por gerar a lista que determina quais clientes poderão contratá-lo. Outro ponto importante é que os espaços de empréstimo/venda são determinados por regras de CRM. Sendo assim, é importante ressaltar que nem todos os empréstimos estarão disponíveis na home. Diante disso, vemos que existem diversas áreas e relações fortes envolvidas na contratação de um empréstimo. Caso isso não seja controlado, negativados, por exemplo, poderiam contratar empréstimos com valores elevados. Em nossa discussão gostaríamos de trazer um exemplo real, na qual um cliente negativo viu um empréstimo aprovado no valor de 105mil reais, porém ele não conseguia efetivá-lo, posteriormente o momento tirou uma print-screen e levou ao BACEN que fez o banco efetivar o empréstimo ao cliente.

O quarto caso é muito semelhante à funcionalidade de extrato, em que o cliente consegue visualizar a lista de transações, fazer filtros dos últimos 15 dias, por exemplo, e obter demais informações. Como falado no segundo caso, transações precisam ser consistentes quanto à exibição dos dados. Um erro nesse aspecto pode apresentar um saldo incorreto para o cliente, o que pode gerar multas ou consequências indesejadas, levando o usuário a tomar decisões com base em dados errados.

Concluindo, vemos que, nos casos estabelecidos, o mais recomendável é o uso do banco relacional. Em alguns cenários até podemos pensar na utilização de bancos não relacionais, porém, no nosso entendimento, preferimos utilizar esse tipo de banco (relacional) para garantir consistência, rastreabilidade e integridade das informações. Já os bancos não relacionais preferimos utilizar no nosso data lake, onde os dados não estruturados ou semi-estruturados podem ser armazenados de forma mais flexível, com foco maior em volume e análise exploratória, e não tanto na transação em si.