INSTITUTO DE COMPUTAÇÃO
www.computacao.ufba.br
MATA60 - Banco de Dados
Prof. Robespierre Pita
robespierre.pita@ufba.br

Trilha Prática

(individual ou dupla)

1. Modelando a base de dados

1.1. Problema

Alice e Bob são sócios de uma rede de mercearias inteligentes, chamada CompraEsperta. As suas lojas são conteiners adapdatos, climatizados e automatizados que ficam espalhados por condomínios em todas as capitais do país. Cada estabelecimento tem um conjunto produtos que variam entre guloseimas, alimentos prontos (congelados), ingredientes de última hora para receitas comuns, utensílios para o lar, etc. Tudo para prover uma comodidade aos clientes. Não existem funcionários alocados na loja, todos os produtos são marcados com um dispositivo de RFID que, depois de pagos, tem saída liberada do conteiner. O pagamento é feito em terminais de auto-atendimento que são operados pelos próprios clientes. Em seu *core-business*, modelos de inteligência artificial são responsáveis por definir os melhores locais para estabelecer depósitos que vão armazenar os produtos de reposição das lojas de cada cidade. Esta otimização garante uma diminuição substantiva dos custos, permitindo que a CompraEsperta consiga repôr rapidamente os estoques de cada loja e estabeleça uma proximidade com os fornecedores que ofereçam preços mais competitivos. Os depósitos contam com uma área de armazenamento maior, garantindo que a empresa compre insumos em grande quantidade e negocie os valores com seus fornecedores.

No último ano, o número médio de contêiners por capital subiu de 1 para 4.6, tornando proibitivo o controle do estoque através de planilhas. Alice e Bob decidiram então contratar uma empresa de engenharia de software para construir um Sistema de Informação sob medida. Este sistema será responsável por controlar o estoque dos contêiners e depósitos. Quando uma loja atinge o estoque mínimo de um conjunto de produtos, ou quando o depósito está desabastecido, o sistema dispara automaticamente ordens de compra para fornecedores ou de reposição para funcionários da logística. Os principais requisitos do sistema são expostos a seguir.

1.2. Requisitos do Sistema de Informação

RF1: O sistema deve controlar o estoque geral da empresa. Desta forma, a distinção entre depósitos ou lojas deve estar determinada em cada localização de um determinado item/produto. Ou seja, os produtos em depósito ou lojas são determinados por um atributo de localização.

RF2: O controle do estoque é feito a partir de produtos individuais ou porções. Apesar da compra ser feita por lotes e caixas, o sistema deve considerar apenas unidades em seus registros e relatórios.

RF3: Cada produto individual está associado a uma *tag* RFID. O status desta tag determinará se um produto ainda está disponível em estoque ou não.

RF4: O sistema deve ser capaz de controlar os fornecedores de cada produto. Multiplas empresas podem fornecer um ou mais items. Fornecedores cuja a data de ultima compra seja maior de 180 dias deve ser considerada inativa.

RF5: O sistema deve ser capaz de prover relatórios que informem: i) o nível de estoque para cada categoria de produtos em cada loja ou depósito; ii) os fornecedores mais frequentes, com menores custos para cada categoria de produtos; iii) as lojas com maior vazão de produtos num período; iv) produtos com maior proximidade de vencimento; etc.

RF6: Cada categoria de produtos tem estoque mínimo e maximo associado. Estes números são distintos para lojas e depósitos.

`No decorrer do processo de desenvolvimento do sistema, novos requisitos podem surgir.`

1.3 Delimitação do mini-mundo para o banco de dados

O banco de dados relacional para suportar as operações deste sistema de informação deve contar com as seguintes entidades e atributos:

- **tbl_produto**: tabela que concentra informações sobre os itens individuais que compõem o acervo/estoque da empresa.
    - **cp_id_produto** [int, incremental]: código identificador do produto individal. Único e incremental.
    - **nm_prod** [str, 60 caracteres]: nome do produto, conforme normas técnicas para integração em notas fiscais (Norma Técnica 2021.004 – v.133).
    - **cd_ean_prod** [str, 12 caracteres]: código de barras do produto, conforme padrão EAN.
    - **ce_rfid** [int, 8 bytes]: chave estrangeira que define o rfid do produto.
    - **ce_categoria_principal** [int, 8 bytes]: chave estrangeira que define a categoria principal do produto.
    - **ce_categoria_secundaria** [int, 8 bytes]: chave estrangeira que define a categoria secundária do produto.
- **tbl_rfid**: tabela que concentra informações sobre os dispositivos de identificação individual dos produtos.

    - **cp_id_dispositivo** [int, 8 bytes]: chave primária que identifica dispositivo.
    - **ind_venda_dispositivo** [bool, 1 byte]: flag que indica se produto já foi vendido (1 = vendido, 0 = em estoque).
- **tbl_categoria**: tabela que concentra informações sobre as categorias de cada produto.
    - **cp_cod_categoria** [int, 4 bytes]: chave primária que identifica a categoria de cada produto.
    - **nm_categoria** [str, 20 caracteres]: nome da categoria.
- **tbl_estabelecimento**: tabela que concentra informações dos estabelecimentos (lojas e depósitos)
    - **cp_cod_estab** [int, 8 bytes]: chave primária que identifica o estabelecimento.
    - **nm_estab** [str, 60 caracteres]: nome do estebelecimento.
    - **cnpj_estab** [str, 60 caracteres]: CNPJ do estebelecimento, em caso de filial ou empresa parceira.
    - **localizacao_estab** [float vector, 8 espaços]: vetor com latitude e longitude dos 4 pontos que delimitam o espaço do estabelecimento no espaço geográfico.
    - **endereco_estab** [str, 200 caracteres]: endereço estebelecimento.
    - **UF_estab** [str, 2 caracteres]: codigo IBGE da UF do estebelecimento.
    - **cidade_estab** [str, 5 caracteres]: codigo IBGE da cidade do estebelecimento.
- **tbl_funcionario**: tabela que concentra informações sobre os funcionários responsáveis pela reposição do estoque nas lojas e depósitos.
    - **cp_cod_func** [int, 8 bytes]: chave primária que identifica o funcionário.
    - **nm_func** [srt, 200 caracteres]: nome completo do funcionário.
    - **cpf_func** [srt, 11 caracteres]: CPF do funcionário.
    - **funcao_func** [srt, 40 caracteres]: função do funcionário.
- **tbl_fornecedor**: tabela que concentra informações sobre os fornecedores dos produtos individuais.
    - **cp_cod_forn** [int, 8 bytes]: chave primaria que identifica o fornecedor.
    - **cnpj_forn** [str, 14 bytes]: cnpj do fornecedor.
    - **localizacao_forn** [float vector, 8 espaços]: vetor com latitude e longitude dos 4 pontos que delimitam o espaço do fornecedor no espaço geográfico.
    - **endereco_forn** [str, 200 caracteres]: endereço fornecedor.
    - **UF_forn** [str, 2 caracteres]: codigo IBGE da UF do fornecedor.
    - **cidade_forn** [str, 5 caracteres]: codigo IBGE da cidade do fornecedor.

As relações entre estas entidades deve seguir as regras a seguir:

- Um fornecedor pode fornecer nenhum ou vários produtos. Produtos podem ser fornecidos por mais de um fornecedor. As relações entre fornecedores e produtos deve gerar informações sobre o preço de venda, data de venda e data de vencimento.
- Um produto só pode ser identificado por um único RFID. RFID são consumidos por apenas um produto.

- Uma mesma categoria pode rotular mais de um produto. Um produto pode ter apenas uma categoria principal e outra secundária.
- Um estabelecimento vende ou distribui vários produtos e um mesmo produto pode ser oferecido por vários estabelecimentos. As relações entre estabelecimento s e produtos deve gerar informações sobre itens comprados, preços e data de venda e outros atributos pertinentes.

Trilha prática 1

A partir da compreensão obtida do Sistema de Informação descrito acima:

1. Sugira novas tabelas, variáveis, relacionamentos ou requisitos;
2. Utilize o BrModelo para criar os modelos conceitual e lógico do banco de dados;
3. Crie o SQL DDL do banco de dados modelado;
4. Popule as tabelas com pelo menos 200 registros.

Trilha prática 2

De acordo com os requisitos:

1. Produza 20 queries básicas, 15 queries intermediárias e 10 queries avançadas;
2. Usando as configurações iniciais:
    a. Execute este conjunto de 45 queries pelo menos 50 vezes;
    b. Colete o tempo de execução de cada querie a cada rodada;
    c. Monte uma planilha de baseline com este tempo.
3. Elabore um plano de indexação e:
    a. Execute este conjunto de 45 queries pelo menos 50 vezes;
    b. Colete o tempo de execução de cada querie a cada rodada;
    c. Monte uma planilha de melhoria de desempenho evidenciando a diferença nos tempos de execução (speedup).
4. Elabore um plano de tunning e:
    a. Execute este conjunto de 45 queries pelo menos 50 vezes;
    b. Colete o tempo de execução de cada querie a cada rodada;
    c. Monte uma planilha de melhoria de desempenho evidenciando a diferença nos tempos de execução (speedup).

Trilha prática 3

INSTITUTO DE COMPUTAÇÃO
www.computacao.ufba.br
MATA60 - Banco de Dados
Prof. Robespierre Pita
robespierre.pita@ufba.br

De acordo com os requisitos, elabore:

1. Duas views;
2. Duas stored procedures;
3. Duas transações.

Sobre a complexidade das queries

CCB - Consultas de complexidade baixa
- Usando ANSI SQL (para garantir padronização com os SGBDs da disciplina), escreva consultas de complexidade baixa para responder questões específicas nas bases de dados.
- A complexidade baixa da consulta está relacionada com um baixo número de tabelas envolvidas (até duas), com listagens ou contagens simples.
- Acompanhada de cada consulta, uma pergunta deve ser fornecida para capturar o que está sendo recuperado da base de dados e suas tabelas.

CCI - Consultas de complexidade intermediária
- Usando ANSI SQL (para garantir padronização com os SGBDs da disciplina), escreva consultas de complexidade baixa para responder questões específicas nas bases de dados.
- A complexidade intermediária da consulta está relacionada com um moderado número de tabelas envolvidas (entre duas e quatro), com operações de junção (JOIN), união (UNION), paginação (WINDOW), agrupamento (GROUPBY) **OU** sub-selects mais simples.
- Acompanhada de cada consulta, uma pergunta deve ser fornecida para capturar o que está sendo recuperado da base de dados e suas tabelas.

CCI - Consultas de complexidade alta
- Usando ANSI SQL (para garantir padronização com os SGBDs da disciplina), escreva consultas de complexidade baixa para responder questões específicas nas bases de dados.
- A complexidade intermediária da consulta está relacionada com um moderado número de tabelas envolvidas (acima de duas), com operações de junção (JOIN), união (UNION), paginação (WINDOW), agrupamento (GROUPBY) **E** sub-selects acompanhadas de ordenação por uma ou duas colunas.
- Acompanhada de cada consulta, uma pergunta deve ser fornecida para capturar

Quadro 1. Barema de avaliação - Trilha Prática 1

Item	Peso	Avaliação		
P1. Construção e avaliação das consultas	6	[Q1] = Criou e avaliou 45 consultas	[Q2] = Q1 + Criou e avaliou corretamente vinte CCB,	[Q3] = Q2 + consultas estão coerentes e relacionadas com os requisitos do sistema.

			quinze CCI; e dez CCA*.	
P2. Modelagem e implantação	4	[Q1] = Modelagem conceitual e lógica estão adequadas		[Q2] = Q1 + E DDL é funcional e materializa a tradução adequada do modelo conceitual para o físico E Questões de uso ótimo do recurso (ex.: armazenamento, memória e indexação) foram endereçadas. E Base populada corretamente com código funcional.
nota		0 | 1 | 2 | 3 | 4		5 | 6 | 7 | 8 | 9 | 10

Quadro 2. Barema de avaliação - Trilha Prática 2

Item	**Peso**	**Avaliação**		
P1. Construção e avaliação das consultas	4	[Q1] = Criou e avaliou 15 consultas	[Q2] = Q1 + Criou e avaliou corretamente cinco CCB, cinco CCI; e cinco CCA*.	[Q3] = Q2 + aprimorou corretamente as 15 consultas avaliadas e avaliou a melhoria usando speedup.
P2. Plano de tuning	4	[Q1] = Construiu os três cenários.		[Q2] = Q1 + E Cenários são válidos (variáveis existem e serviço continua funcionando) E Cenários são compostos de, pelo menos, 5 variáveis E Equipe executou todas as consultas de forma automatizada, coletando os resultados diretamente da execução.

P3. Plano de indexação	2	[Q1] = Construiu os três cenários.					[Q2] = Q1 + E Cenários são válidos (conjuntos diferentes: variável OU técnicas de indexação) E Discente usou pelo menos uma variável em cada tabela E Discente executou todas as consultas de forma automatizada, coletando os resultados diretamente da execução. E Discente executou todas as consultas de forma automatizada, coletando os resultados diretamente da execução.					
nota		0	1	2	3	4	5	6	7	8	9	10

Quadro 3. Barema de avaliação - Trilha Prática 3

Item	**Peso**	**Avaliação**										
P1. Views ou materialized Views	3	[Q1] = Criou duas views ou materialized views funcionais					[Q3] = Q2 + Atende pelo menos um dos requisitos do SI					
P2. Stored Procedures	3	[Q1] = Criou duas stored procedures					[Q3] = Q2 + Atende pelo menos um dos requisitos do SI					
P2. Transações	4	[Q1] = Criou duas transações					[Q3] = Q2 + Atende pelo menos um dos requisitos do SI					
nota		0	1	2	3	4	5	6	7	8	9	10