Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Disciplina: Laboratório de Banco de Dados
Profª Dra. Fátima Nunes.

Administração de Condomínio

Parte II

Fernando K. G. de Amorim – 10387644
João Guilherme da Costa Seike – 9784634
Lucas Pereira Castelo Branco – 10258772
Victor Gomes de O. M. Nicola – 9844881

São Paulo
2019

LETRA C

Especificação do Requisito

Para esse requisito, levantou-se a necessidade de ter-se a relação das pessoas e seus bens dentro do condomínio. É importante ressaltar a importância do sigilo desses dados entre os envolvidos, visto a questão da privacidade de dados tão presente. Para tal, podemos definir formalmente:

"O sistema deverá ser capaz de permitir a filiais interessadas que haja uma relação dos condôminos que estão cadastrados dentro de condomínios ligados a sua administração, correlacionando com todos os bens que estão sob sua custódia."

Solução em SQL Padrão:

```
CREATE VIEW relatorio_pessoas_filial AS
	SELECT  pessoa.nome AS nome,
				pessoa.cpf AS cpf,
				(CASE WHEN moradia.tipo_moradia = 'c'
					THEN 'Casa'
					  WHEN moradia.tipo_moradia = 'a'
					THEN 'Apartamento'
					ELSE NULL
					END
				 ) AS tipo_moradia,
				(CASE WHEN tipo_moradia = 'Casa'
					THEN casa.numero_casa
					ELSE  apartamento.numero_ap --Assumindo que toda moradia que não é uma casa, é um apartamento
					END
				) AS numero,
				apartamento.andar,
				apartamento.final,
				veiculo.placa,
				veiculo.marca,
				veiculo.modelo--,
				--COALESCE(condominio_moradia.fk_id_condominio, edificio.fk_id_condominio)
				--	AS final_fk_id_condominio --Auxiliar para nossa query abaixo

				FROM adm_condominio.pessoa AS pessoa

				--Dando JOIN na parte de moradia
				JOIN adm_condominio.moradia_pessoa AS moradia_pessoa
				  ON moradia_pessoa.fk_id_pessoa = pessoa.id_pessoa
				JOIN adm_condominio.moradia AS moradia
				  ON moradia.id_moradia = moradia_pessoa.fk_id_moradia
				 --Temos LEFT JOIN aqui, pois a pessoa só terá um deles
		LEFT JOIN adm_condominio.casa AS casa
				  ON casa.fk_id_moradia = moradia.id_moradia
		LEFT JOIN adm_condominio.apartamento AS apartamento
				  ON apartamento.fk_id_moradia = moradia.id_moradia
```

```
                        -- Bloco de JOINs para conseguirmos testar o condominio e a filial na qual ele
pertence
                       JOIN adm_condominio.condominio_moradia AS condominio_moradia
                        ON condominio_moradia.fk_id_moradia = moradia.id_moradia
                       JOIN adm_condominio.condominio AS condominio
                        ON condominio.id_condominio  = condominio_moradia.fk_id_condominio
                       JOIN adm_condominio.condominio_filial AS condominio_filial
                        ON condominio_filial.fk_id_condominio = condominio.id_condominio

                        -- Temos LEFT JOIN porque a pessoa pode não ter veiculo
        LEFT JOIN adm_condominio.veiculo_moradia AS veiculo_moradia
                        ON veiculo_moradia.fk_id_moradia = moradia.id_moradia
        LEFT JOIN adm_condominio.veiculo AS veiculo
                        ON veiculo.id_veiculo = veiculo_moradia.fk_id_veiculo

                       -- LIMITADOR DE SEGURANÇA: vemos apenas a lista de pessoas ligada a essa
filial
                       WHERE condominio_filial.fk_id_filial = 5
```

Solução em código implementada

```sql
CREATE VIEW relatorio_pessoas_filial AS
    SELECT  pessoa.nome AS nome,
            pessoa.cpf AS cpf,
            (CASE WHEN moradia.tipo_moradia = 'c'
                THEN 'Casa'
                  WHEN moradia.tipo_moradia = 'a'
                THEN 'Apartamento'
                ELSE NULL
                END
             ) AS tipo_moradia,
            (CASE WHEN tipo_moradia = 'Casa'
                THEN casa.numero_casa
                ELSE apartamento.numero_ap --Assumindo que toda moradia que não é uma casa, é um apartamento
                END
            ) AS numero,
            apartamento.andar,
            apartamento.final,
            veiculo.placa,
            veiculo.marca,
            veiculo.modelo--,
            --COALESCE(condominio_moradia.fk_id_condominio, edificio.fk_id_condominio)
            --  AS final_fk_id_condominio --Auxiliar para nossa query abaixo

            FROM adm_condominio.pessoa AS pessoa

            --Dando JOIN na parte de moradia
            JOIN adm_condominio.moradia_pessoa AS moradia_pessoa
              ON moradia_pessoa.fk_id_pessoa = pessoa.id_pessoa
            JOIN adm_condominio.moradia AS moradia
              ON moradia.id_moradia = moradia_pessoa.fk_id_moradia
             --Temos LEFT JOIN aqui, pois a pessoa só terá um deles
        LEFT JOIN adm_condominio.casa AS casa
              ON casa.fk_id_moradia = moradia.id_moradia
        LEFT JOIN adm_condominio.apartamento AS apartamento
              ON apartamento.fk_id_moradia = moradia.id_moradia

              -- Bloco de JOINs para conseguirmos testar o condominio e a filial na qual ele pertence
            JOIN adm_condominio.condominio_moradia AS condominio_moradia
              ON condominio_moradia.fk_id_moradia = moradia.id_moradia
            JOIN adm_condominio.condominio AS condominio
              ON condominio.id_condominio  = condominio_moradia.fk_id_condominio
            JOIN adm_condominio.condominio_filial AS condominio_filial
              ON condominio_filial.fk_id_condominio = condominio.id_condominio

             -- Temos LEFT JOIN porque a pessoa pode não ter veiculo
        LEFT JOIN adm_condominio.veiculo_moradia AS veiculo_moradia
              ON veiculo_moradia.fk_id_moradia = moradia.id_moradia
        LEFT JOIN adm_condominio.veiculo AS veiculo
             ON veiculo.id_veiculo = veiculo_moradia.fk_id_veiculo

            -- LIMITADOR DE SEGURANÇA: vemos apenas a lista de pessoas ligada a essa filial
            WHERE condominio_filial.fk_id_filial = 5
```

Transcrição do código

A ideia é que, a partir dessa lista, consigamos fazer toda a lista dos itens que estão cadastrados no sistema que estão ligados a pessoa, ou seja, a sua moradia e seu veículo.
Para que isso fosse possível, as três tabelas básicas usadas foram ‘pessoa’, ‘moradia’ e ‘veículo’. Porém, para a restrição, foi utilizado também a tabela ‘condomínio’.

Durante a etapa de selecionar os campos, fazemos um tratamento simples para que o campo se torne mais legível ao usuário final. Podemos observar este caso em ‘tipo_moradia”, onde fazemos uma leve conversão de acordo com o caractere registrado. Outro tratamento realizado para garantir que haja uma tabela mais legível é a existência de um campo flexível: em ‘número”, realizamos um teste justamente com o tipo de moradia para permitir que o campo tenha significado de acordo com o contexto da linha.

Já na parte de agrupar as tabelas, podemos dividir em quatro blocos, como explicado no primeiro parágrafo. Na primeira parte, pegamos como base a tabela ‘pessoa’. A partir dele, a segunda parte realiza um INNER JOIN com ‘moradia’, pois apenas teremos registros da pessoa se necessariamente ele

possuir uma moradia. Porém, no conceito de ‘casa’ e ‘apartamento’, pode-se dizer que uma pessoa muito possivelmente terá apenas um deles, sendo assim necessário um LEFT JOIN para garantir a funcionalidade.
Por fim, no bloco usado para a filtragem, usamos de JOIN para permitir uma ligação entre a moradia e o ‘condomínio’ na qual está presente.

Para essa VIEW em específico, assumiu-se que a Filial de número 5 no sistema fora a solicitadora principal dessa possibilidade, e portanto, o WHERE se baseia nele. Para fins práticos, assumiu-se que apenas ela será consultada.

Custo-Benefício e Atualização de Dados

O ganho em custo-benefício é evidente. A complexidade de montar uma query nesse formato mostra quão prático pode ser termos uma tabela pronta, que pode ser acionada em uma operação SELECT simples. Quando contamos com a presença de um modelo que possui sentido prático para o usuário final dentro de seu caso de uso, o fato de permitirmos uma visão pré-estabelecida facilita sua construção a longo prazo. Em questão de funcionalidade, também permite com que tenhamos variações dessa view para outros, já que sua função é garantir que cada filial tenha visão apenas de seus próprios condôminos.
A atualização de dados não é tão constante quanto à consulta, mas sua importância é inegável: a filial deve manter uma relação de seus condôminos, tanto por questões financeiras (garantindo que todos estão pagos) como para questões legais (que podem ir desde a questão de ter-se os dados das pessoas em casos de crimes até verificação de quebras de contrato).