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 D

Especificação do Requisito

Uma das consultas mais comuns em bancos de dados como esse é justamente o gerenciamento do fluxo de entrada e saída aos condomínios. Por isso, é interessante haver alguma forma de já termos, condensado em uma tabela, alguma forma de visualizarmos isso de forma mais abrangente, de forma a permitir sua manipulação quando necessário.
Para isso, contamos com três tabelas: 'entrada_saida' é a principal, responsável por identificar o momento da ação e seu tipo; e o que chamaremos de ator, que pode ser uma pessoa ou um veículo.
O requisito, formalizado, será:
"O sistema deve ser capaz de encontrar, com uma latência de menos de um segundo, os registros de fluxo de condôminos, de forma a permitir não apenas identificá-los, como também verificar quais as últimas ocorrências na qual o sistema tem ciência."

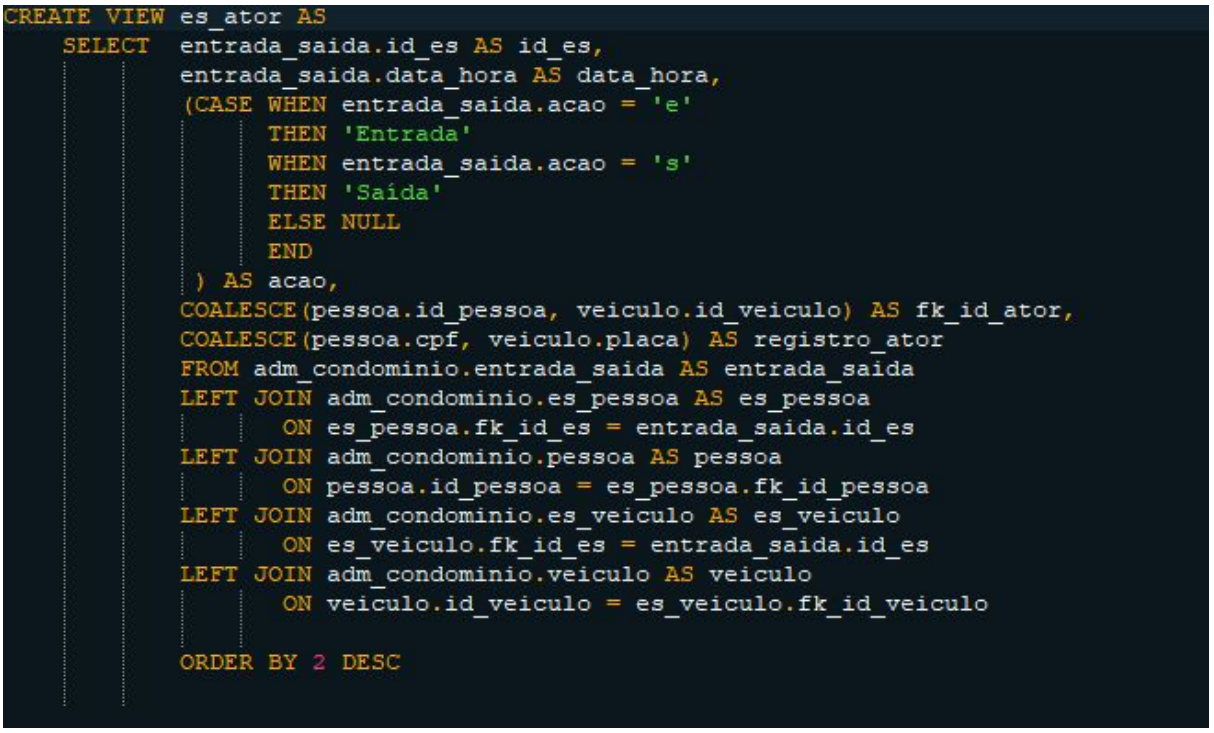
Solução em SQL Padrão:

```
CREATE VIEW es_ator AS
        SELECT   entrada_saida.id_es AS id_es,
                        entrada_saida.data_hora AS data_hora,
                        (CASE WHEN entrada_saida.acao = 'e'
                                THEN 'Entrada'
                                WHEN entrada_saida.acao = 's'
                                THEN 'Saída'
                                ELSE NULL
                                END
                         ) AS acao,
                        COALESCE(pessoa.id_pessoa, veiculo.id_veiculo) AS fk_id_ator,
                        COALESCE(pessoa.cpf, veiculo.placa) AS registro_ator
                        FROM adm_condominio.entrada_saida AS entrada_saida
                        LEFT JOIN adm_condominio.es_pessoa AS es_pessoa
                                ON es_pessoa.fk_id_es = entrada_saida.id_es
                        LEFT JOIN adm_condominio.pessoa AS pessoa
                                ON pessoa.id_pessoa = es_pessoa.fk_id_pessoa
                        LEFT JOIN adm_condominio.es_veiculo AS es_veiculo
                                ON es_veiculo.fk_id_es = entrada_saida.id_es
                        LEFT JOIN adm_condominio.veiculo AS veiculo
                                ON veiculo.id_veiculo = es_veiculo.fk_id_veiculo

                        ORDER BY 2 DESC
```

Solução em código implementada

Para tal caso, temos a seguinte visão:



```
CREATE VIEW es_ator AS
    SELECT  entrada_saida.id_es AS id_es,
            entrada_saida.data_hora AS data_hora,
            (CASE WHEN entrada_saida.acao = 'e'
                  THEN 'Entrada'
                  WHEN entrada_saida.acao = 's'
                  THEN 'Saída'
                  ELSE NULL
                  END
             ) AS acao,
            COALESCE(pessoa.id_pessoa, veiculo.id_veiculo) AS fk_id_ator,
            COALESCE(pessoa.cpf, veiculo.placa) AS registro_ator
            FROM adm_condominio.entrada_saida AS entrada_saida
            LEFT JOIN adm_condominio.es_pessoa AS es_pessoa
                  ON es_pessoa.fk_id_es = entrada_saida.id_es
            LEFT JOIN adm_condominio.pessoa AS pessoa
                  ON pessoa.id_pessoa = es_pessoa.fk_id_pessoa
            LEFT JOIN adm_condominio.es_veiculo AS es_veiculo
                  ON es_veiculo.fk_id_es = entrada_saida.id_es
            LEFT JOIN adm_condominio.veiculo AS veiculo
                  ON veiculo.id_veiculo = es_veiculo.fk_id_veiculo

            ORDER BY 2 DESC
```

Transcrição do código

Neste código, optamos por duas características: a presença de identificadores únicos ao sistema (visto sua utilidade posterior em consultas); e a aglutinação de informações relevantes ao usuário, de forma a permitir uma maior facilidade tanto para o mantenedor do banco quanto para o usuário final.
Para isso, criamos uma ‘entidade’, que é chamada ‘ator’, que representa quem de fato está ligado a aquela ação de entrada/saída, podendo ser uma pessoa ou um veículo. Dessa forma, usamos da função COALESCE, que irá obter a primeira informação não-nula do sistema, e a partir disso, exibir a informação relevante a aquele registro.

Custo-Benefício e Atualização de Dados

O custo-benefício é simples: a partir dessa visão, já temos a junção de informações simples, porém essenciais para a segurança do condomínio. Além disso, se quisermos eventualmente obter mais informações sobre a pessoa, ou sobre a moradia, por exemplo, podemos fazê-lo usando do ‘id_ator’ para fazermos os JOINs necessários e completar a consulta. Dessa forma, conseguimos atender de antemão ao anseio básico do administrador como também auxiliamos a construção de consultas mais complexas ao banco, em caso de necessidade.

Alguns exemplos de queries que podemos utilizar nesse caso são:

- *Query que realiza a listagem de todas as entradas e saídas e seus respectivos atores, considerando apenas aquelas que ocorreram antes do período de 2 dias atrás.*

```
SELECT *
FROM es_ator
    WHERE data_hora < NOW() - interval '2 days'
```

Para essa query, usamos do recurso NOW(), que registra o horário atual do sistema na qual o SGBD está inserido. A partir de 'interval' permite com que façamos operações com ela.

- *Conta o número de registros de entrada e saída por tipo de moradia.*

```
SELECT   moradia.tipo_moradia,
                    COUNT(es_ator.id_es)
FROM es_ator
LEFT JOIN adm_condominio.veiculo_moradia AS veiculo_moradia
          ON es_ator.fk_id_ator = veiculo_moradia.fk_id_veiculo
LEFT JOIN adm_condominio.moradia_pessoa AS moradia_pessoa
          ON es_ator.fk_id_ator = moradia_pessoa.fk_id_pessoa
JOIN adm_condominio.moradia AS moradia
          ON moradia.id_moradia IN (veiculo_moradia.fk_id_moradia, moradia_pessoa.fk_id_moradia)
        GROUP BY 1
```

Nessa query, aproveitamos do fato de 'ator' poder representar duas entidades para fazer uma contagem total ligada a outra entidade, 'moradia', que possui relacionamento tanto com 'pessoa' quanto com 'veiculo'.

A questão da atualização de dados é latente. A tabela de entrada e saída tende a ser a mais modificada do banco, ao se tratar de um registro de atividades dos condomínios. Portanto, a presença de uma visão como essa privilegia que haja um acompanhamento *real-time* dessa atualização e seus pares pertinentes (no caso, o envolvido na ação em questão). Por isso, não é uma boa ideia torná-la uma view persistente no banco, visto sua volatilidade constante.