Projeto Integrador – Banco de Dados 2

Professor: Jônatas Costa de Oliveira

Equipe: Édson Santarém Bastos

Isabelle Blanco Lima Loureiro

Jamielson da Silva Magalhães

Rafael Santos Alves

Resolução – Sistema de Condominio

1. [1,0 ponto] Criação de, no mínimo, 3 consultas complexas (com junções, filtros e/ou agrupamentos e funções de agregação).
   1. Nome dos moradores que reservaram local e não possuem empréstimo.

SELECT M.MORADOR\_NOME FROM MORADOR AS M

INNER JOIN RESERVA\_DE\_AREA\_COMUM AS RAC

ON RAC.MORADOR\_COD= M.MORADOR\_COD

WHERE NOT EXISTS (SELECT E.MORADOR\_COD FROM EMPRESTIMO AS E

WHERE E. MORADOR\_COD = M.MORADOR\_COD);

* 1. Listar o(s) Nome(s) do(s) objeto(s) que foi(ram) emprestado(s) por moradores que possuem multas em ordem alfabética.

SELECT E.EMPRESTIMO\_ITEM FROM EMPRESTIMO AS E

INNER JOIN MORADOR AS MO

ON MO.MORADOR\_COD= E.MORADOR\_COD

INNER JOIN MULTA AS MU

ON MU.MORADOR\_COD= MO.MORADOR\_COD

ORDER BY E.EMPRESTIMO\_ITEM ASC

* 1. Nome do morador que possui o maior valor de multa no condomínio e que não fez reserva de área comum.

SELECT M.MORADOR\_NOME FROM MORADOR AS MO

INNER JOIN MULTA AS MU

ON MU.MORADOR\_COD= MO.MORADOR\_COD

AND MAX (MULTA\_VALOR)

WHERE NOT EXISTS (SELECT RAC.MORADOR\_COD FROM RESERVA\_DE\_AREA\_COMUM AS RAC

WHERE RAC.MORADOR\_COD= MO.MORADOR\_COD);

1. [1,0 ponto] Criação de, no mínimo, 4 índices.
   1. Create Index ind\_Multa\_Morador On Multa (multa\_Valor, morador\_cod);
   2. Create Index Ind\_Ocorrencia\_Feita On Ocorrencia (ocorrencia\_data, ocorrencia\_desc, ocorrencia\_local, morador\_cod);
   3. Create Index Ind\_Visitas On Visita (Visita\_Nome, Cod\_Morador);
   4. Create Index Ind\_Morador\_Ativo On Morador (Morador\_Nome) Where morador\_situacao =”Ativo”;
2. [1,5 pontos] Criação de, no mínimo, 2 visões.
   1. Create View view\_area\_comum\_ocupada

(Codigo, descricao, Situacao)

as Select area\_cod, area\_desc, area\_situacao

From AreaComum

Where area\_situacao =”Ocupada”

* 1. Create View view\_encomendas\_entregues

(Código\_Encomenda, Data\_Entrada, Data\_Recebida, Morador, Imovel)

as Select encomenda\_cod as codigo, encomenda\_data\_chegada, encomenda\_data\_retirada, morador\_cod, imovel\_cod

From Encomendas

Where encomenda\_data\_retirada is not null

1. [1,0 ponto] Criação de, no mínimo, 1 trigger.
2. [1,5 pontos] Criação de, no mínimo, 2 funções ou stored procedures.
   1. Uma função que faz a contagem de moradores com multas.

CREATE OR REPLACE FUNCTION QNT\_MORADOR\_MULTA ()

RETURNS INTEGER AS $$

DECLARE

QNTD INTEGER;

BEGIN

SELECT COUNT (\*)

INTO QNTD

FROM MORADOR MO

INNER JOIN MULTA MU

ON MO.MORADOR\_COD = MU.MORADOR\_COD

AND MU.MULTA\_VALOR IS NOT NULL;

RETURN QNTD;

END;

$$ LANGUAGE PLPGSQL;

* 1. Uma função que faz a contagem de empréstimo realizada em determinado dia. (obs. : o dia deve ser informado para que seja retornado algo.

CREATE OR REPLACE FUNCTION QNT\_EMPRESTIMO\_DIA (DIA EMPRESTIMO. emprestimo\_Data%TYPE) RETURNS INTEGER AS $$

DECLARE

QNTD INTEGER;

BEGIN

SELECT COUNT (\*)

INTO QNTD

FROM EMPRESTIMO E

WHERE DIA = E. emprestimo\_Data;

RETURN QNTD;

END;

$$ LANGUAGE PLPGSQL;

1. [1,0 ponto] Criação de, no mínimo, 1 transação.
   1. Uma transação que faz o empréstimo de um item e a reserva de uma área do condomínio

BEGIN;

INSERT INTO EMPRESTIMO (emprestimo\_data, emprestimo\_Data\_Devolucao, emprestimo\_Item, morador\_cod, imovel\_cod) VALUES ('12/03/2018', '14/03/2018',‘2’, '68', '454');

INSERT INTO RESERVA\_AREA\_COMUM (reserva\_Cod, reserva\_Data\_Inicial, reserva\_Data\_Final, area\_Cod, morador\_Cod) VALUES ('354', '12/03/2018', '14/03/2018'',‘8’, '2');

COMMIT;

END;