Exercicio 1:

Antes da criação dos procedimentos.

Uma imagem com captura de ecrã, Software de multimédia, software

Os conteúdos gerados por IA poderão estar incorretos.

Figura - tabela reservas

Uma imagem com captura de ecrã, texto, preto

Os conteúdos gerados por IA poderão estar incorretos.

Figura - tabela pagamentos

Código da criação dos procedimentos:

CREATE OR REPLACE PROCEDURE inserir\_reserva (r\_id\_cliente INT, r\_id\_quarto INT, r\_data\_checkin DATE, r\_data\_checkout DATE)

LANGUAGE plpgsql

AS $$

DECLARE

    r\_id\_reserva INT; *-- variavel para guardar o id da reserva para depois associar ao pagamento*

BEGIN

*-- Inserir reserva com estado 'pendente'*

    INSERT INTO RESERVAS (ID\_UTILIZADORES, ID\_QUARTOS, DATA\_CHECKIN, DATA\_CHECKOUT, ESTADO\_RESERVA, VALOR\_TOTAL, DATA\_CRIACAO)

    VALUES (r\_id\_cliente, r\_id\_quarto , r\_data\_checkin, r\_data\_checkout,'pendente', (SELECT preco FROM QUARTOS WHERE ID\_QUARTOS = r\_id\_quarto) ,now()) RETURNING ID\_RESERVAS INTO r\_id\_reserva;

    call inserir\_pagamento(r\_id\_reserva, 'multibanco');

END;

$$;

CREATE OR REPLACE PROCEDURE inserir\_pagamento(p\_id\_reserva INT, p\_metodo VARCHAR(50))

LANGUAGE plpgsql

AS $$

DECLARE

    p\_id\_pagamento INT; *-- variavel para guardar o id do pagamento para depois associar a reserva*

BEGIN

    INSERT INTO PAGAMENTOS (ID\_RESERVAS, VALOR\_PAGO, METODO, DATA\_PAGAMENTO, ESTADO\_PAGAMENTO)

    VALUES (p\_id\_reserva, (SELECT valor\_total FROM RESERVAS WHERE ID\_RESERVAS = p\_id\_reserva), p\_metodo, now(), 'pago') RETURNING ID\_PAGAMENTOS INTO p\_id\_pagamento;

    UPDATE RESERVAS SET ESTADO\_RESERVA = 'confirmada', ID\_PAGAMENTOS = p\_id\_pagamento WHERE ID\_RESERVAS = p\_id\_reserva;

END;

$$;

Chamada dos procedimentos para inserir reservas e pagamentos:

CALL inserir\_reserva(3, 1, '2021-06-01', '2021-06-05');

CALL inserir\_reserva(3, 2, '2021-06-01', '2021-06-05');

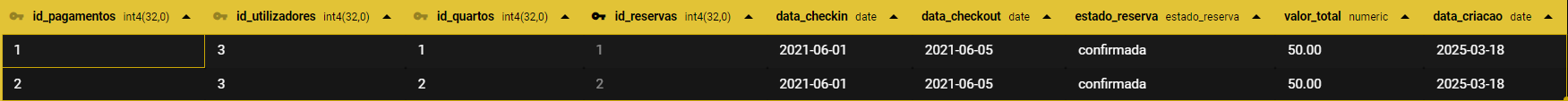
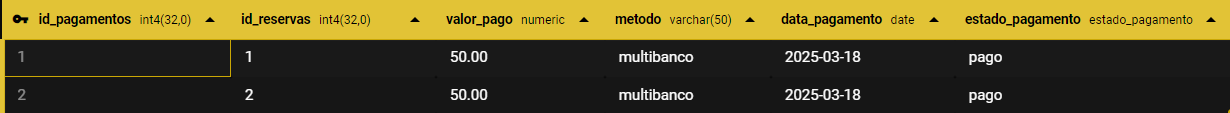
Pós execução:

Figura - pós procedimento tabela pagamentos

Figura - pós procedimento reservas

Exercicio 2:

create or replace function is\_available(num\_quarto int, data date)

returns boolean as $result$

declare

    result boolean;

begin

    select exists(

        select quartos.id\_quartos

        from quartos, reservas

        where quartos.id\_quartos = reservas.id\_quartos

            and data between reservas.data\_checkin and reservas.data\_checkout

            and quartos.numero = num\_quarto

    ) into result;

    return not result;

end;

$result$ language plpgsql;

Execução:

SELECT is\_available(101, '2025-02-04')

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Os conteúdos gerados por IA poderão estar incorretos.

Figura - pós função com um valor válido

SELECT is\_available(101, '2021-06-03')

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Os conteúdos gerados por IA poderão estar incorretos.

Figura - pós função com um valor inválido

Exercicio 3:

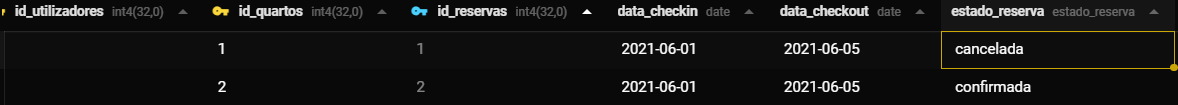
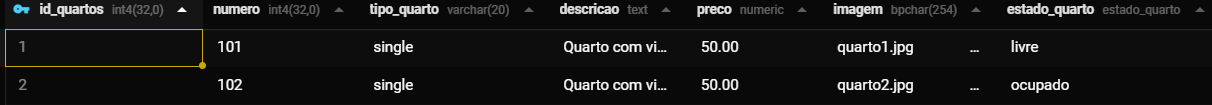
Codigo:

Figura - pós criação do trigger tabela quartos

Figura - pos criação do trigger tabela reservas

create or replace function status\_trigger\_func()

returns trigger as $status\_trigger$

begin

    if new.estado\_reserva = 'confirmada' then

        update quartos

            set estado\_quarto = 'ocupado'

            where id\_quartos = new.id\_quartos;

    elseif new.estado\_reserva = 'cancelada' then

        update quartos

            set estado\_quarto = 'livre'

            where id\_quartos = new.id\_quartos;

    end if;

    return new;

end;

$status\_trigger$ language plpgsql;

create trigger status\_trigger

after insert or update on reservas

for each row

execute procedure status\_trigger\_func();

Os exercício 1 e 2 funcionam na sua totalidade, o exercício 3 altera corretamente o status do quarto diretamente associado a reserva no entanto não alterar qualquer outro quarto.