

# Desafio PA - SQL

## Davi Dias Monsores dos Santos

Neste documento serão explicadas as estratégias de implementação utilizadas para mexer com os dados contidos no banco de dados. Uma observação geral importante é que para casos onde a reversão dos dados seria complexa ou impossível após implementações de atualização ou remoção fossem realizadas, foi utilizado “Begin”, “Commit” e “End”, apenas chamando o “Commit” após verificar se os casos estavam corretos, modificando os dados como deveria.

### **1- Inserir Novo Cliente: Cadastre um novo cliente chamado "Tech4Bank" no sistema (tabela CLIENT).**

**R:** Após observar a tabela CLIENT e suas colunas de criação, vi que os atributos eram os seguintes:

- id: Chave primária, do tipo uuid;
- name: Nome do cliente, do tipo varchar; e
- created\_at: Momento de criação do cliente, do tipo timestamp without timezone.

Sabendo disso, na hora de realizar a inserção, solicitei que o ChatGPT criasse um uuid aleatório para mim, preenchi o name com o pedido no desafio e o created\_at com o momento no qual criei o novo cliente.

Código SQL:

```
insert into "CLIENT"  
values ('3f1d7a92-5c03-4f09-8bfa-efba2e26334f', 'Tech4Bank', '2025-05-22 21:02:50')
```

### **2- Adicionar Serviço ao Cliente: Associe um novo serviço chamado "Consultoria Financeira Personalizada" à cliente "Tech4Bank" (tabela CLIENT\_SERVICE).**

**R:** Após observar a tabela CLIENT\_SERVICE e suas colunas de criação, vi que os atributos eram os seguintes:

- id: Chave primária, do tipo uuid;
- client\_id: Chave estrangeira, do tipo uuid;
- service\_name: Nome do serviço a ser vinculado com o cliente, do tipo varchar; e
- created\_at: Momento de criação do serviço do cliente, do tipo timestamp without timezone.

Sabendo disso, para realizar a inserção dos dados, mais uma vez solicitei que o ChatGPT criasse um uuid aleatório para ser o id. Além disso, coloquei o id antes inserido para o cliente Tech4Bank como a chave estrangeira client\_id, para que fizesse referência ao id do cliente na tabela CLIENT. Por fim, fiz a inserção do serviço solicitado no desafio no service\_name e o created\_at com o momento no qual criei o novo serviço de cliente.

Código SQL:

```
insert into "CLIENT_SERVICE"
values ('f7b05d24-b0af-4906-9327-63c00c4cf7f0',
       '3f1d7a92-5c03-4f09-8bfa-efba2e26334f',
       'Consultoria Financeira Personalizada',
       '2025-05-22 21:34:25')
```

**3- Criar Novo Formulário: Crie um novo formulário (FORMS) com o nome "Declaração de Bens" e descrição "Formulário para declaração de patrimônio do solicitante".**

**R:** Após observar a tabela FORMS e suas colunas de criação, vi que os atributos eram os seguintes:

- id: Chave primária, do tipo uuid;
- name: Nome do formulário, do tipo varchar;
- description: Descrição do formulário, do tipo text; e
- created\_at: Momento de criação do formulário, do tipo timestamp without timezone.

Sendo assim, no momento de realizar a inserção eu solicitei mais um uuid aleatório para o ChatGPT, para inserir no id. Após isso, fiz a inserção do name e da description conforme pedido no desafio e, por fim, coloquei o momento da criação do formulário no created\_at.

Código SQL:

```
insert into "FORMS"
values ('b989fa89-6e7f-426a-8890-c8dd888bb785',
       'Declaração de Bens',
       'Formulário para declaração de patrimônio do solicitante',
       '2025-05-22 21:51:02')
```

**4- Adicionar Campos ao Formulário:**

- Ao formulário "Declaração de Bens", adicione os seguintes campos (FORM\_FIELD):
  - label: "Descrição do Bem", type: "text", required: true
  - label: "Valor Estimado do Bem", type: "number", required: true
  - label: "Possui Alienação?", type: "boolean", required: false

**R:** Após observar a tabela FORM\_FIELD e suas colunas de criação, vi que os atributos eram os seguintes:

- id: Chave primária, do tipo uuid;
- form\_id: Chave estrangeira fazendo referência ao id de um formulário, do tipo uuid;
- label: Campo do formulário, do tipo varchar;
- type: Tipo do campo do formulário, do tipo varchar;
- required: Diz se o campo é obrigatório no formulário, do tipo boolean;

- validation: Diz se é necessária alguma validação em relação ao preenchimento do campo, do tipo text; e
- created\_at: Momento de criação do campo do formulário, do tipo timestamp without timezone.

Dessa forma, pedi ao ChatGPT para criar 3 novos uuid aleatórios para a inserção de cada um dos campos de formulário. Após isso, fiz a inserção do id do formulário criado anteriormente no form\_id, para ser a chave estrangeira referenciada. Em seguida, realizei a inserção dos valores pedidos pelo desafio em label, type e required. Ao ver que em nenhum deles foi necessário validação, deixei a coluna em branco, já que ela não possui not null. Por fim, coloquei o momento da criação dos 3 campos de formulário.

Código SQL:

```
insert into "FORM_FIELD" (id, form_id, label, type, required, created_at)
values ('edca7854-90c8-4c93-bdef-71e9e2ac9420',
       'b989fa89-6e7f-426a-8890-c8dd888bb785',
       'Descrição do Bem',
       'text',
       true,
       '2025-05-22 22:12:23'),
       ('05cb7d3c-efc0-4f5a-bd3d-e236ecdbf59a',
       'b989fa89-6e7f-426a-8890-c8dd888bb785',
       'Valor Estimado do Bem',
       'number',
       true,
       '2025-05-22 22:12:23'),
       ('0f3692f5-5d56-43c7-9aa4-f1e5e75bcb0c',
       'b989fa89-6e7f-426a-8890-c8dd888bb785',
       'Possui Alienação?',
       'boolean',
       false,
       '2025-05-22 22:12:23')
```

**5- Atualizar Dados do Cliente: O cliente com \_id (um ID fictício que você pode definir ou buscar, ex: 15) mudou de nome para "Tech4BankEditado". Atualize o registro na tabela CLIENT.**

**R:** Para atualizar o id de um cliente baseado em seu id, utilizei a instrução update para atualizar a tabela “CLIENT”, defini o novo atributo name, conforme pedido no desafio, apenas para o cliente com id igual ao id do cliente criado anteriormente.

Código SQL:

```
update "CLIENT"
set name = 'Tech4BankEditado'
where id = '3f1d7a92-5c03-4f09-8bfa-efba2e26334f'
```

**6- Modificar Campo de Formulário: No formulário "Dados Pessoais", altere o campo "CPF" (label: "CPF") para que sua validação (validation) agora exija um formato específico (ex: regex para ###.###.###-##). Se a coluna validation não existir ou precisar ser adaptada, considere como faria isso.**

**R:** Para modificar algo de um campo de formulário a partir do nome do formulário, utilizei um select para buscar o id do formulário em questão na tabela "FORMS", já que o id do forms é algo que a tabela "FORM\_FIELD" possui. Sendo assim, atualizei, onde o label fosse CPF e o id do forms igual ao id do forms "Dados Pessoais", o validation da tabela para um regex que pede como entrada 3 números, um ponto, mais 3 números, outro ponto, mais 3 números, um hífen e, por fim, dois números. Caso não existisse a coluna validation eu teria que criá-la por meio de:

```
alter table "FORM_FIELD"  
add column validation text;
```

Código SQL:

```
update "FORM_FIELD"  
set validation = '^\\d{3}\\.\\d{3}\\.\\d{3}-\\d{2}$'  
where label = 'CPF' and  
form_id = (select id from "FORMS"  
           where name = 'Dados Pessoais')
```

**7- Atualizar Ordem de Passo em Workflow: Em um workflow existente (ex: workflow com \_id 20), altere a ordem (order) do WORKFLOW\_STEP "Formulário de Dados de Endereço" de 2 para 3, e ajuste o passo que anteriormente era 3 para 2.**

**R:** Para trocar os passos 2 e 3 do workflow que possuía o passo "Formulário de Dados de Endereço", foi necessário utilizar uma expressão case, onde os passos 3 viram 2 e os passos 2 viram 3. Além disso, para que não fossem alterados os passos de outros workflows, utilizei o select para filtrar essas alterações somente para quem tiver o mesmo workflow\_id do Formulário de Dados de Endereço. Uma observação é que o order precisou ser entre aspas pelo comando order já estar presente de outra maneira no SQL.

Código SQL:

```
update "WORKFLOW_STEP"  
set "order" = case "order"  
    when 3 then 2  
    when 2 then 3  
    else "order"  
end  
where workflow_id = (select workflow_id from "WORKFLOW_STEP"  
                    where step_name = 'Formulário de Dados de Endereço');
```

**8- Remover Opção de Campo:** De um campo do tipo select ou radio (ex: campo "Estado Civil" no formulário "Dados Pessoais"), remova uma opção específica (ex: "Viúvo(a)") da tabela FORM\_OPTIONS.

**R:** Para remover uma opção do campo de um formulário, eu precisei aninhar duas consultas utilizando o select. A primeira foi para remover apenas a opção do campo onde o id fosse do campo Estado Civil e a segunda foi para remover apenas a opção do campo de Estado Civil que fizesse parte do formulário de Dados Pessoais. Dessa forma, foi possível remover a opção "Solteiro(a)" do formulário.

Código SQL:

```
delete from "FORM_OPTIONS"
where label = 'Solteiro(a)'
and field_id = (select id from "FORM_FIELD"
                where label = 'Estado Civil'
                and form_id = (select id from "FORMS"
                               where name = 'Dados Pessoais'));
```

**9- Excluir um Serviço de Cliente:** Remova o serviço "Seguro Residencial" de um cliente específico (identificado pelo client\_id e service\_name na tabela CLIENT\_SERVICE).

**R:** Para remover o serviço de Seguro Residencial de um cliente em específico, utilizei o select para verificar se o id do cliente salvo no serviço seria igual ao id do cliente específico decidido, nesse caso "Cliente Exemplo Antigo". Dessa forma, apenas o serviço deste cliente em específico é apagado.

Código SQL:

```
delete from "CLIENT_SERVICE"
where service_name = 'Seguro Residencial'
and client_id = (select id from "CLIENT"
                 where name = 'Cliente Exemplo Antigo');
```

**10- Desativar/Excluir Workflow:** Um workflow chamado "Análise de Crédito Simplificada" tornou-se obsoleto.

- Primeiro, delete todos os seus WORKFLOW\_STEP associados.
- Em seguida, delete o próprio WORKFLOW.

**R:** Para resolver esse caso, pensei em duas possibilidades. Uma delas é apenas remover todos os WORKFLOW\_STEP com workflow\_id igual ao id do WORKFLOW de Análise de Crédito Simplificada e depois remover o WORKFLOW com nome de Análise de Crédito Simplificada. A outra possibilidade seria com a inserção da constraint "on delete cascade" na chave estrangeira, primeiro removendo a constraint de chave estrangeira da tabela com:

```
alter table "WORKFLOW_STEP"  
drop constraint nome_da_constraint;
```

E depois inserindo novamente, agora com a cascata:

```
alter table "WORKFLOW_STEP"  
add constraint nome_da_constraint  
foreign key (workflow_id) references "WORKFLOW"(id) on delete cascade;
```

Ambas as formas são funcionais, mas escolhi implementar a primeira por ser a de mais simples implementação.

Código SQL:

```
delete from "WORKFLOW_STEP"  
where workflow_id = (select id from "WORKFLOW"  
                     where name = 'Análise de Crédito Simplificada');  
  
delete from "WORKFLOW"  
where name = 'Análise de Crédito Simplificada';
```