Escopo por unidade

Vincule a ação ao recurso correto

Pré-requisitos

Conheça papéis de acesso e consultas SQL. O exercício usa Python 3 e SQLite. A conta já foi autenticada e cada reserva pertence a uma unidade. A gestão do exemplo não inclui permissão de check-in.

Use uma atribuição com contexto [1]

Uma conta pode atuar na recepção do Centro e na gestão do Sul. Registre conta, unidade e papel no mesmo vínculo. Ao verificar uma operação, busque um vínculo único que inclua o papel requerido, a unidade do recurso e a condição de vigência. Avaliar papel e unidade em listas independentes perde a associação entre eles.

Obtenha a unidade do registro

Ao receber o ID de uma reserva, determine a unidade pelo registro persistido. A unidade apresentada no seletor da interface pode servir como filtro de navegação. A decisão do servidor usa o estado carregado do banco e o identificador da conta que a autenticação estabeleceu.

Use a regra em leituras e escritas [1][2]

Para listar reservas internas, filtre na consulta as unidades nas quais a conta tem o papel pertinente. Para mudar um registro, inclua o escopo na própria operação ou aplique uma verificação equivalente com isolamento adequado. EXISTS testa se há pelo menos uma linha que satisfaça todos os critérios. Uma escrita condicional também pode exigir o estado de origem.

Cuide de dados e resultados

A contagem zero indica que a escrita não ocorreu. Ela pode resultar de falta de vínculo, ausência da reserva ou estado não elegível. Defina respostas que orientem quem pode ver a reserva e preservem informações de recursos fora de seu escopo. Atribuições revogadas precisam deixar de ser consideradas pela política.

Escopo por unidade

Exercício resolvido • Check-in no Centro ou no Sul

Enunciado e dados de partida

Centro é a unidade 1 e Sul é a unidade 2. A conta 12 atende no Centro e é gestora no Sul. Seu vínculo antigo de recepção no Sul está inativo. A conta 77 não tem vínculos. R81 é do Centro e R82 é do Sul. As duas começam confirmadas.

1. Salve dados.sql em UTF-8

```
CREATE TABLE vinculos (
  conta_id INTEGER NOT NULL,
  unidade_id INTEGER NOT NULL,
  papel TEXT NOT NULL,
  ativo INTEGER NOT NULL CHECK (ativo IN (0, 1)),
  PRIMARY KEY (conta_id, unidade_id, papel)
);
CREATE TABLE reservas (
  id INTEGER PRIMARY KEY,
  unidade_id INTEGER NOT NULL,
  estado TEXT NOT NULL
);

INSERT INTO vinculos VALUES
  (12, 1, 'recepcao', 1),
  (12, 2, 'gestao', 1),
  (12, 2, 'recepcao', 0);
INSERT INTO reservas VALUES
  (81, 1, 'confirmada'),
  (82, 2, 'confirmada');
```

2. Salve checkin.sql

```
UPDATE reservas
SET estado = 'presente'
WHERE id = :reserva
  AND estado = 'confirmada'
  AND EXISTS (
    SELECT 1
    FROM vinculos v
    WHERE v.conta_id = :conta_autenticada
      AND v.unidade_id = reservas.unidade_id
      AND v.papel = 'recepcao'
      AND v.ativo = 1
  );
```

Explique a condição

A consulta interna compara a unidade do vínculo à unidade da linha que seria atualizada. O mesmo vínculo precisa atender às outras condições. O UPDATE exige reserva confirmada, o que impede registrar a chegada novamente após a mudança para presente.

Escopo por unidade

Execute e confira quais linhas mudaram

3. Salve executar.py e rode python executar.py

```
from pathlib import Path
import sqlite3

banco = sqlite3.connect(':memory:')
banco.executescript(Path('dados.sql').read_text(encoding='utf-8'))
sql = Path('checkin.sql').read_text(encoding='utf-8')

def checkin(conta, reserva):
    args = {'conta_autenticada': conta, 'reserva': reserva}
    return banco.execute(sql, args).rowcount

assert checkin(77, 81) == 0
assert checkin(12, 82) == 0
assert checkin(12, 81) == 1
assert checkin(12, 81) == 0

estados = banco.execute('SELECT estado FROM reservas ORDER BY id')
assert [estado for (estado,) in estados] == ['presente', 'confirmada']
print('Apenas a reserva autorizada mudou');
```

Leia os resultados

A conta 77 não altera R81. A conta 12 também não altera R82, pois ali a recepção está inativa. A mesma conta altera R81 exatamente uma vez. O teste usa dados simulados e avalia o efeito no banco. A abrangência de transações concorrentes e revogação em produção depende do banco e da política adotados.

Variação e critérios

Ative a recepção no Sul para a conta 12 antes das chamadas e ajuste a segunda asserção para esperar uma linha e o estado final de R82 para presente. R81 continua seguindo a regra do Centro. Execute cada versão com dados iniciais novos. Confira que o acesso ao Sul depende da atribuição naquele local.

Referências primárias

[1] OWASP • Validação de permissões a cada requisição

[2] PostgreSQL • Semântica da expressão EXISTS

Escopo por unidade

Carregue a unidade junto dos dados solicitados

Delimite também as consultas indiretas

Filtros por unidade devem chegar às consultas de listagem, detalhe e exportação. No detalhe, busque a reserva com seu ID e a unidade autorizada. Aplique o mesmo recorte aos itens relacionados.

Exercício adicional

O relatório mostra apenas reservas da unidade Centro, mas inclui anexos da unidade Sul. O que corrigir?

Resolução

Aplique o escopo também na leitura dos anexos e valide o vínculo deles com as reservas selecionadas. Faça um teste com dados fictícios de ambas as unidades.

Ao assistir ao vídeo

Identifique onde o isolamento é aplicado no desenho do serviço.

Vídeo complementar

AWS re:Invent - Multi-tenancy no AWS Fargate (em inglês)