PROCEDURES
(Procedimentos)

FAPAM

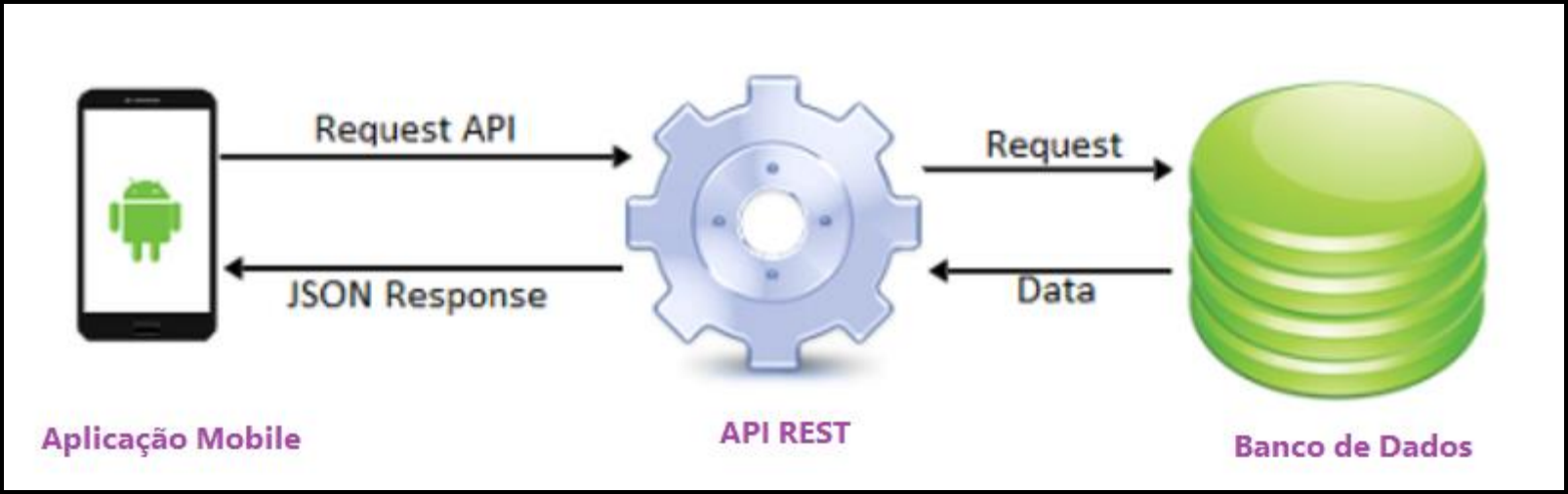
PROCEDURES

- O que é um Stored Procedure?
- São rotinas definidas no banco de dados, identificadas por um nome pelo qual podem ser invocadas.

- Um procedimento pode executar uma série de instruções e receber parâmetros.

PROCEDURES

- Para que usar uma Stored Procedure?
- Muitas vezes é requerido várias consultas e atualizações no BD, o que acarreta um maior consumo de recursos pela aplicação (desempenho, memória, etc). No caso de aplicações web, isso se torna mais visível, devido a maior quantidade de informações que precisam trafegar pela rede e de requisições ao servidor.



PROCEDURES

- Uma boa forma de contornar ou atenuar esse consumo de recursos diretamente pela aplicação é transferir parte do processamento para o BD. Assim, considerando que as máquinas servidoras geralmente têm configurações de hardware mais robustas (e nada se pode garantir com relação às máquinas clientes), essa pode ser uma “saída” a se considerar.

PROCEDURES

- USAR OU NÃO USAR PROCEDURES?
- Como exemplo para o funcionamento dos Stored Procedures, iremos comparar a execução de uma rotina utilizando e outra não utilizando essa técnica.
- Considere o seguinte contexto de uma aplicação comercial:
- O cliente faz um pedido, no qual são inseridos itens.
- O pedido (bem como os itens) permanecem com status “PENDENTE” até ser confirmado.
- O operador confirma o pedido e faz o registro no livro caixa.

PROCEDURES

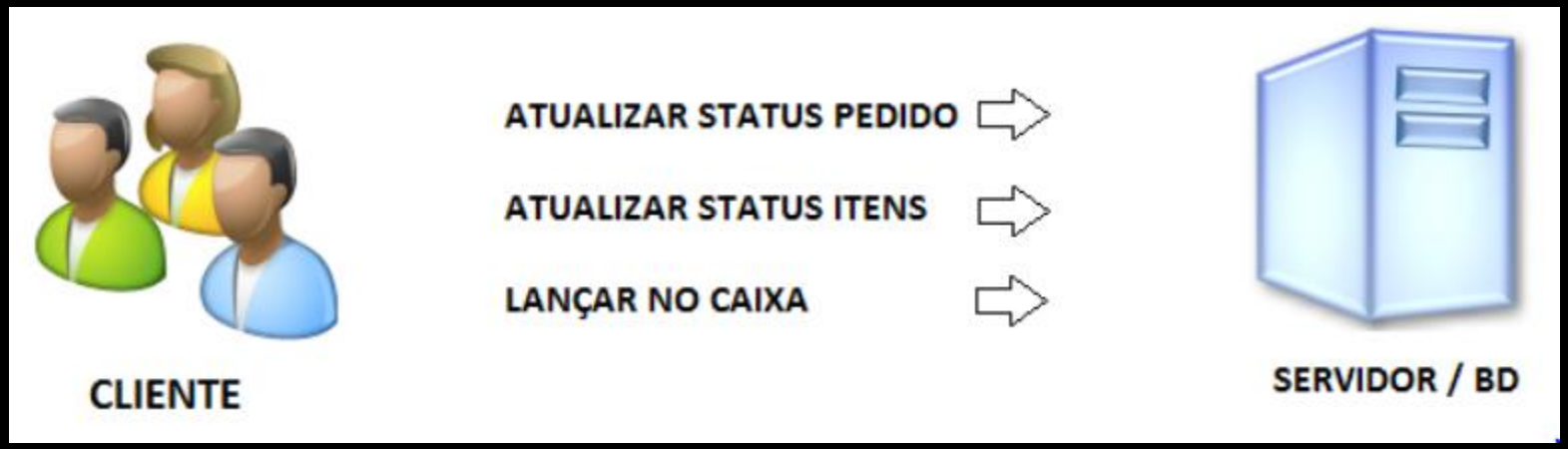
- USAR OU NÃO USAR PROCEDURES?
- Até o pedido ser confirmado, nenhum lançamento é feito no livro caixa, então é preciso ter uma rotina de confirmação do pedido, que deve executar as seguintes ações:

- Atualizar o status do pedido (fechado, pendente) – UPDATE
- Atualizar o status dos itens do pedido (vendido, pendente) - UPDATE
- Lançar o valor do pedido no caixa (preço) - INSERT

PROCEDURES

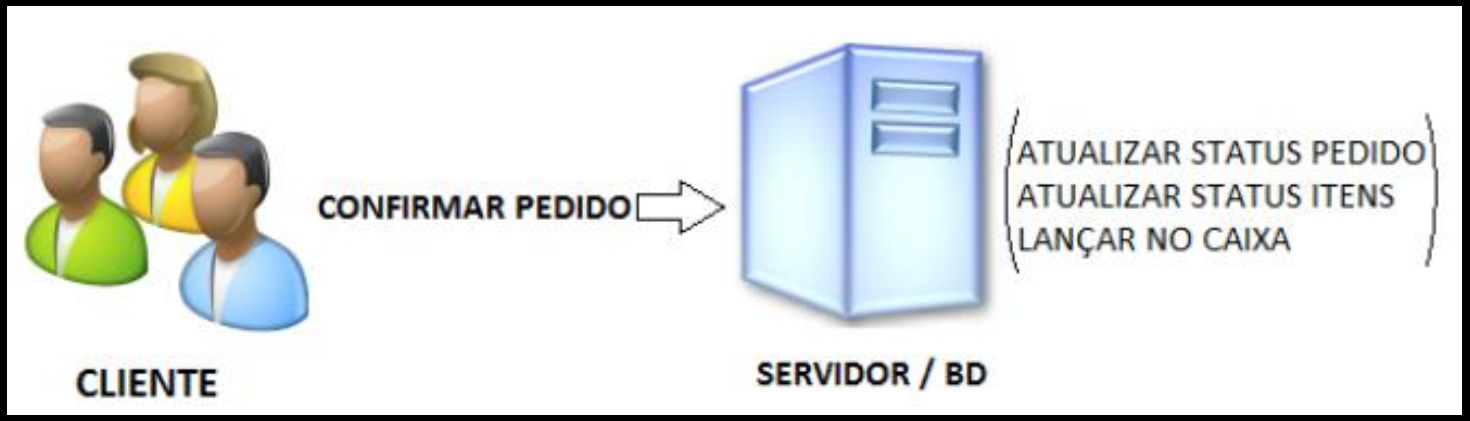
- USAR OU NÃO USAR PROCEDURES?

- Temos então pelo menos 3 instruções de atualização e/ou inserção.
- Poderíamos representar essa situação graficamente pela Figura 1.



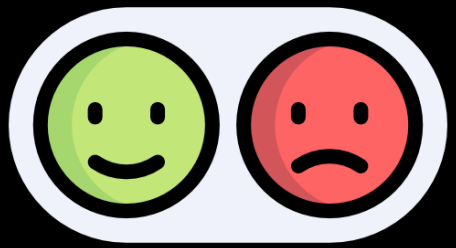
PROCEDURES

- USAR OU NÃO USAR PROCEDURES?
- Por outro lado, poderíamos agrupar essas três instruções no corpo de um procedimento e chamá-lo a partir da aplicação uma única vez.
- As ações de UPDATE/INSERT/DELETE, a partir daí, ficariam por conta do servidor. A representação gráfica desse modelo é mostrada a seguir (através do procedimento chamado “CONFIRMAR PEDIDO”).



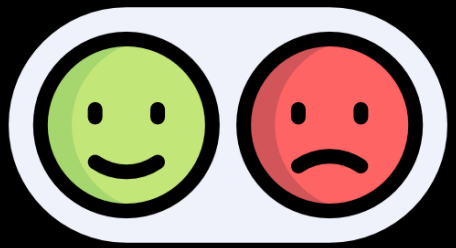
PROCEDURES

- USAR OU NÃO USAR PROCEDURES?
- VANTAGENS:
- Simplificação da execução de instruções SQL pela aplicação.
- Transferência de parte da responsabilidade de processamento para o servidor.
- Facilidade na manutenção, reduzindo a quantidade de alterações na aplicação.



PROCEDURES

- DESVANTAGENS:
- Necessidade de maior conhecimento da sintaxe do banco de dados para escrita de rotinas em SQL (nível avançado).
- As rotinas ficam mais facilmente acessíveis. Alguém que tenha acesso ao banco de dados poderá visualizar e alterar o código.



PROCEDURES

- Sintaxe:
- CREATE PROCEDURE insert_data(a integer, b integer)
- LANGUAGE SQL
- AS $$
- INSERT INTO tbl VALUES (a);
- INSERT INTO tbl VALUES (b);
- $$;
- nome_procedimento: o nome que identificará o procedimento (o que quiser).
- parâmetros: são opcionais e, caso não sejam necessários, devem permanecer apenas os parênteses vazios na declaração da procedure.

PROCEDURES

- Exemplo:

```
CREATE PROCEDURE insert_cliente(nome
VARCHAR(30), cpf VARCHAR(11), celular VARCHAR(11))
LANGUAGE SQL
AS $$
INSERT INTO cliente VALUES
((SELECT MAX(codigo_cliente) + 1 FROM cliente), nome,
cpf, celular);
$$;
```

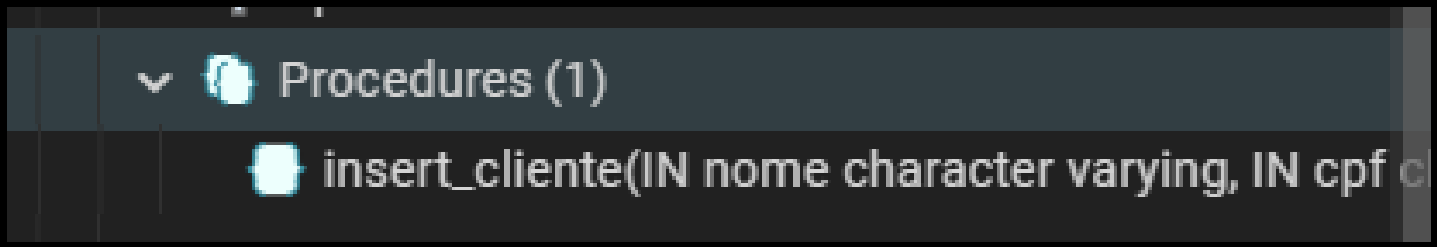
PROCEDURES

- Exemplo:

```
CREATE PROCEDURE insert_cliente(nome
VARCHAR(30), cpf VARCHAR(11), celular VARCHAR(11))
LANGUAGE SQL
AS $$
INSERT INTO cliente VALUES
((SELECT MAX(codigo_cliente) + 1 FROM cliente), nome,
cpf, celular);
$$;
```

PROCEDURES

- No pgAdmin



PROCEDURES

- Tendo criado o procedure, chamá-lo é bastante simples.
- Para isso fazemos uso da palavra reservada CALL, como mostra o código a seguir:

- Sintaxe para chamar um STORED PROCEDURE:

CALL “nome_procedimento”(lista_parametros);

PROCEDURES

- Exemplo:

```
CALL insert_cliente('Gabriel', '11111111111', '37999999999');
```

PROCEDURES

- Atualizando uma procedure:
- Para atualizar o código de uma procedure, basta usar o comando CREATE OR REPLACE PROCEDURE
- Exemplo:

```
CREATE OR REPLACE PROCEDURE
update_preco(novo_preco NUMERIC, codigo_produto
INTEGER)
LANGUAGE SQL
AS $$
UPDATE produto SET preco = novo_preco WHERE codigo
= codigo_produto;
$$;
```

PROCEDURES

- Assim como outras estruturas no banco de dados, para exclusão de procedures basta fazer:

- Sintaxe: DROP PROCEDURE ;

- Exemplo:
- DROP PROCEDURE Elevar_Ao_Quadrado;
- DROP PROCEDURE Verificar_Quantidade_Produtos;