Como os dados são processados

O método `getPedidosRecebidos()` executa um único comando SQL complexo que usa **CTEs** (Common Table Expressions — subconsultas nomeadas) para organizar os dados em etapas.

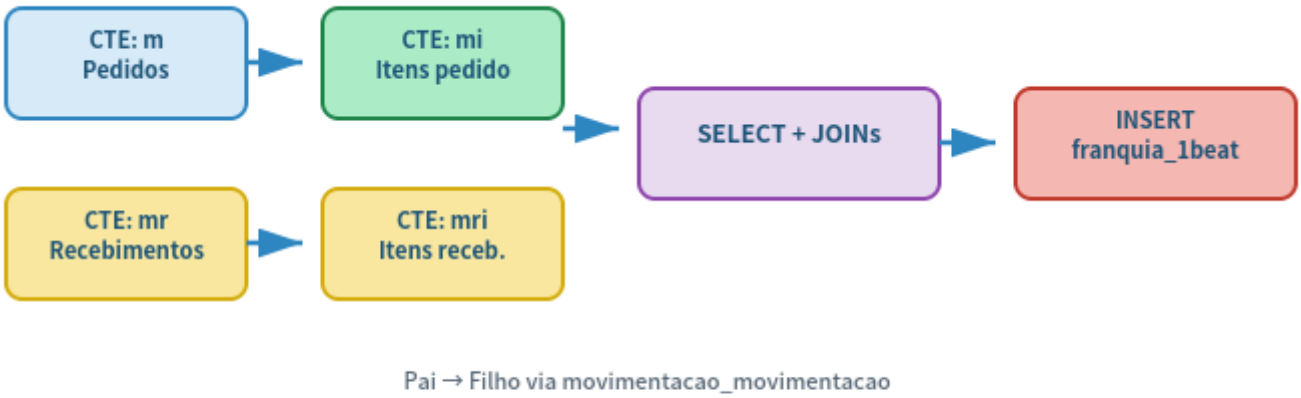


Figura 9 — As CTEs organizam pedidos, itens e recebimentos antes do INSERT final.

Etapa por etapa

1. CTE `m` — Pedidos no período

Seleciona movimentações do tipo PEDIDO alteradas entre o ponteiro e a data limite. Faz LEFT JOIN com `stage_pedido_lx` para integração LX.

2. CTE `mi` — Itens do pedido

Para cada pedido, soma as quantidades de cada produto (grade) nos itens ativos do pedido. Também captura o flag `pedido_corte_item` (usado no cálculo de QTDE_CORTADA e QTDE_PENDENTE).

3. CTE `mr` — Recebimentos vinculados

Busca movimentações filhas (recebimentos) ligadas ao pedido pai, apenas com situação `Finalizado`.

4. CTE `mri` — Itens recebidos

Soma as quantidades recebidas de cada produto em cada recebimento vinculado ao pedido.

5. SELECT final — Montagem do registro

Junta pedido + itens pedidos + itens recebidos + dados da filial (nome, CNPJ, código).

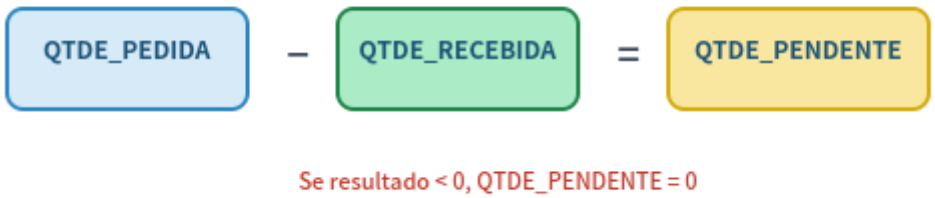


Figura 10 — Cálculo de pendente e cortada: depende do flag `pedido_corte_item` no item.

Campos calculados

Campo destino	Como é calculado
`QTDE_PEDIDA`	Soma dos itens do pedido (`mi.qtde`)
`QTDE_RECEBIDA`	Soma dos itens recebidos (`COALESCE(mri.qtde, 0)`)
`QTDE_CORTADA`	Se `pedido_corte_item = 'S'`: `MAX(0, pedida - recebida)`; senão: `0`
`QTDE_PENDENTE`	Se `pedido_corte_item = 'S'`: `0`; senão: `MAX(0, pedida - recebida)`

`ENTRADA_CONFIRMADA`	`1` se situação = FINALIZADO, senão `0`
`INATIVO`	`1` se situação = Cancelado, senão `0`
`CONFIRMADO`	`0` se Cancelado ou Aberto; `1` nas demais situações
`DATA_EXCLUSAO_MOVIMENTACAO`	`m.data_exclusao` se Cancelado; senão `NULL`
`DATA_TRANSFERENCIA`	`NOW()` no momento da gravação
`SKU`	Código do produto (`mi.codigo`)
`CHAVE_NFE`	Chave da nota fiscal do pedido

Flag `pedido_corte_item`: vem de `movimentacao_detalhe.pedido_corte_item` (prioriza o valor do recebimento, se existir). Quando igual a `'S'`, a diferença entre pedido e recebido vai para `QTDE_CORTADA` em vez de `QTDE_PENDENTE`.

Atualização de registros existentes

O INSERT usa `ON DUPLICATE KEY UPDATE`, ou seja:

- Se o registro **não existe** (chave primária nova) → insere.
- Se o registro **já existe** → atualiza: QTDE_RECEBIDA, QTDE_PEDIDA, QTDE_CORTADA, QTDE_PENDENTE, ENTRADA_CONFIRMADA, DATA_TRANSFERENCIA, INATIVO, CONFIRMADO e DATA_EXCLUSAO_MOVIMENTACAO (esta última só quando INATIVO = 1).

Chave primária: `ID_PRODUTO_GRADE + COD_FILIAL + ID_FILIAL + ID_MOVIMENTACAO_PAI + CNPJ_FILIAL`

Filtros aplicados

Filtro	Valor	Motivo
`modulo`	'PEDIDO'	Apenas movimentações de pedido
`tipo_estoque`	'PEDIDO'	Confirma tipo de estoque
`m.data`	> ponteiro AND <= limite	Janela incremental
`md.situacao`	'ATIVO'	Itens não excluídos
`md.id_produto_grade`	IS NOT NULL	Produtos com grade definida
`mr.situacao`	'Finalizado'	Recebimentos concluídos

Dicionário da coluna CONFIRMADO

Situação no ERP	CONFIRMADO
Cancelado ou Aberto	0
Demais situações (Confirmado, Recebido, Finalizado, etc.)	1

Revision #5
Created 2026-06-24 19:52:49 UTC
Updated 2026-07-09 13:23:15 UTC