

Case - Tabela transacional

Contexto

Imagine-se como um recente integrante da equipe de dados do Pagar.me no papel de Data Developer. Para o seu primeiro desafio, pede-se que implemente uma visão dos dados transacionais que possibilite uma maior facilidade nas consultas por parte das pessoas de negócio e operação, que se queixam sobre a dificuldade de extrair informações relevantes acerca do nosso negócio, devido ao modo como os dados transacionais são salvos.

Considere que todas as informações desejadas possam ser obtidas por meio de quatro tabelas: "Transactions", "Payables", "TransactionOperations" e "Companies". Faça um código que leia os dados das tabelas, processe-os e gere a visão pedida pelo time de negócios.

Os entregáveis deverão ser:

- Arquivo com o código
- Arquivo(s) com a visão final requisitada
- Um texto explicando os passos e premissas adotadas

Algumas dicas para o seu desafio:

- Caso não consiga finalizar a atividade dentro do prazo estipulado, não deixe de enviá-la. Levaremos em consideração tudo o que foi feito e a linha de raciocínio empregada;
- Na escolha da modelagem, adote aquela que irá proporcionar maior facilidade de consulta para as áreas interessadas. Leve em consideração que esta visão poderá ser utilizada por um ferramenta de BI;
- Deixe claras as premissas adotadas. Não levaremos em conta apenas o resultado final, mas também o raciocínio desenvolvido;
- Faça testes entre a sua visão final e as tabelas de origem. Ex: quantidade de transações capturadas por hora, quantidade de transações psp / gateway, quantidade de transações por parcelamento, etc;
- Como material de apoio, não deixe de assistir nosso vídeo sobre Data Warehouse (<https://www.youtube.com/watch?v=yyUsMsD8X2k>);
- Todas as datas nas tabelas de origem estão em UTC;
- Todos os valores (amounts e fees) estão em centavos;
- Nem todos os campos das tabelas de origem serão utilizados. Então, concentre-se nas informações pedidas nos requisitos;
- Em caso de dúvidas, fique a vontade para nos mandar um email :)

Requisitos

Para a realização das análises pedidas pelo time de negócio e operação, é importante criar uma tabela (ou tabelas) que atenda os requisitos listados abaixo.

1. Ter uma linha por id de transação (`transaction_id`) por operação financeira (`financial_operation_type`).
2. Ter os seguintes indicadores numéricos:
 - **transaction_amount**: Valor da transação (em centavos)
 - **mdr_fee**: taxa de mdr da transação para a operação financeira correspondente (em centavos)
 - **gateway_fee**: taxa de gateway da transação (em centavos)
3. Ter apenas transações de cartão (crédito ou débito).
4. Para operações financeiras que indiquem "entrada" de receita (**capture** e **chargeback_refund**), os indicadores **mdr_fee** e **transaction_amount** deverão ser positivos. Para operações financeiras que indiquem "saída" de receita (**refund** e **chargeback**), tais indicadores deverão ser negativos. Assim, se somarmos todas as entradas para uma mesma transação, teremos o valor que será liquidado para o cliente de fato.
5. Considere que não há devolução da taxa de gateway em caso de estorno ou chargeback. Portanto, a taxa de gateway deve ser apenas contabilizada na operação financeira de captura (**capture**), permanecendo zerada para outras operações financeiras.
6. Os atributos de tempo (`created_at`, `date`, `month_name` e `day_of_week`) devem ser preenchidos com a data da operação financeira no horário do Brasil.
7. Ser possível agrupar pelos atributos:
 - **transaction_id** (*varchar*): Chave de identificação da transação;
 - **payment_method** (*varchar*): Meio de pagamento;
 - **installments_range** (*varchar*): Tipo de parcelamento da transação, podendo ser **a vista**, **2-6**, **7-12**;
 - **financial_operation_type** (*varchar*): Tipo da operação financeira (**capture**, **refund**, **chargeback** e **chargeback_refund**);
 - **created_at** (*timestamp*): Horário da ocorrência da operação;
 - **date** (*date*): Data ('YYYY-MM-DD') da ocorrência da operação;
 - **month_name** (*varchar*): Nome do mês da ocorrência da operação. Ex: **Janeiro**, **Fevereiro**, etc;
 - **day_of_week** (*int*): Dia da semana da ocorrência da operação, onde **1** é segunda-feira e **7** é domingo;
 - **company_id** (*varchar*): Chave de identificação do lojista;
 - **company_mcc** (*varchar*): Mcc do lojista;
 - **company_type** (*varchar*): Tipo da afiliação do lojista;
 - **product_name** (*varchar*): Nome do produto utilizado, podendo ser **gateway** ou **psp**;
 - **acquirer_name** (*varchar*): Nome da adquirente que processou a transação;

Descrição das tabelas de origem

Transactions

A tabela "Transactions" apresenta o registro de todas as transações criadas, tanto de cartão quanto boleto. Os campos da tabela são descritos abaixo:

- **transaction_id** (*varchar*): Chave primária da tabela "Transactions". Identifica unicamente uma transação;
- **transaction_created_at** (*timestamptz*): Horário de criação da transação;
- **transaction_updated_at** (*timestamptz*): Horário da última atualização sofrida pela transação;
- **acquirer_name** (*varchar*): Nome do elemento que processou a transação. Quando a transação passa como **gateway**, recebe-se o nome da adquirente que processou a transação. Caso passe como **psp**, o campo é preenchido com **pagarme**;
- **company_id** (*varchar*): Chave primária da tabela "Companies". Identifica unicamente um cliente do Pagar.me;
- **transaction_installments** (*int*): Número de parcelas da transação;
- **amount** (*int*): Valor da transação;
- **payment_method** (*varchar*): Método de pagamento utilizado na compra (cartão de crédito, débito ou boleto);
- **gateway_fee** (*float*): Taxa cobrada pela utilização do gateway Pagar.me. Transações **psp** também são cobradas pela taxa de gateway;
- **status** (*varchar*): Status da transação no momento da consulta;

É importante ressaltar que o atributo **status** sofre atualizações quando ocorre alguma operação relacionada a transação, sempre indicando o estado da transação no momento da consulta. Por exemplo:

Para cartão:

- Quando a transação é paga, vai para o status **paid**
- Quando sofre um estorno, vai para **refunded**
- Quando sofre uma contestação, vai para **chargedback**
- Quando sofre um cancelamento da contestação, volta **paid**

Para boleto:

- Quando o boleto é emitido, vai para o status **waiting_payment**
- Quando é pago, vai para **paid**
- Quando sofre um pedido de estorno inicialmente vai para **pending_refund** e, depois, vai para **refunded**
- Quando um estorno é cancelado, volta para **paid**

Caso a transação não seja autorizada pelo banco emissor ou seja recusada pelo antifraud, vai para o status **refused**.

Consideramos que as transações possuem duas taxas possíveis: taxa de mdm (encontra-se na tabela Payables) e taxa de gateway. A taxa de mdm é cobrada quando o cliente usa o Pagar.me como PSP, neste caso o Pagar.me participa do fluxo do dinheiro. A taxa de gateway é cobrada sempre que participamos do fluxo da informação, seja como gateway ou PSP.

Payables

A tabela "Payables" possui apenas o registro das transações que foram feitas utilizando o Pagar.me como PSP, pois possui as informações acerca do dinheiro que será liquidado para o cliente. Assim, o "payable" (recebível) é o dado de um pagamento referente a uma transação.

A tabela possui uma granularidade dada por transação, parcela, tipo de operação financeira e recebedor (para entender um pouco mais sobre o split de pagamentos, leia a docs <https://docs.pagar.me/docs/overview-marketplace>). Os campos são descritos abaixo:

- **payable_id** (*varchar*): Chave primária da tabela "Payables". Identifica unicamente um payable;
- **payable_created_at** (*timestampz*): Horário de criação do payable;
- **transaction_id** (*varchar*): Chave primária da tabela "Transactions". Identifica unicamente uma transação;
- **company_id** (*varchar*): Chave primária da tabela "Companies". Identifica unicamente uma afiliação;
- **amount** (*int*): Valor da parcela do payable para uma transação e recebedor específico;
- **fee** (*int*): Valor de mdm cobrado, no caso de transações de cartão e valor de conciliação para transações de boleto;
- **installment** (*int*): número da parcela;
- **type** (*varchar*): Tipo da operação financeira (**credit**, **chargeback**, **chargeback_refund** e **refund**). O tipo **credit** é usado para todas as transações de cartão que são capturadas e para todos os boletos que são conciliados;
- **recipient_id** (*varchar*): Chave de identificação de um recebedor de uma transação.
- **payment_method** (*varchar*): Método de pagamento utilizado na compra (cartão de crédito, débito ou boleto).

Obs: Uma possível maneira de se determinar o horário de captura da transação de cartão que passou como **psp** é por meio do horário de criação do primeiro payable do tipo **credit**. De maneira análoga, é possível encontrar a data de criação para as outras operações (**chargeback**, **chargeback_refund** e **refund**).

Companies

A tabela "Companies" contém informações referentes aos clientes cadastrados.

- **company_id** (*varchar*): Chave primária da tabela "Companies". Identifica unicamente um cliente;

- **company_created_at** (*timestampz*): Data de criação da afiliação;
- **company_type** (*varchar*): O tipo da company é uma classificação utilizada pelo time de negócios para diferenciar os clientes (método de prospecção, perfil de cliente, etc);
- **company_mcc** (*varchar*): MCC (Merchant Category Code) do cliente;

TransactionOperations

Quando uma transação é criada, ocorre uma série de operações até que ela chegue a seu estado final. A tabela "TransactionOperations" contém o registro histórico destas operações. À seguir, são descritos os campos da tabela:

- **transaction_operations_id** (*varchar*): Chave primária da tabela "TransactionOperations". Identifica unicamente uma operação da transação;
- **transaction_id** (*varchar*): Chave primária da tabela "Transactions". Identifica unicamente uma transação;
- **status** (*varchar*): Estado da operação. Os valores possíveis: **waiting**, **processing**, **failed**, **success**, **dropped**. Caso uma transação seja negada na autorização, a linha de authorize como **failed** e a linha de captura ficará como **dropped**.
- **type** (*varchar*): Tipo da operação. As operações existentes são:
 - **authorize**: Operação feita para qualquer tipo de transação (cartão ou boleto) com a finalidade de autorizar o valor da transação junto ao banco emissor;
 - **capture**: Operação feita apenas para transações de cartão com a finalidade de confirmar a cobrança autorizada junto ao banco emissor;
 - **issue**: Operação realizada apenas para boletos com o intuito de realizar a emissão do boleto (código de barras e url);
 - **conciliate**: Operação que identifica o pagamento de boletos e chargebacks;
 - **refund**: Operação de estorno, válida para qualquer tipo de transação;
- **acquirer_name** (*varchar*): Nome do elemento que processou a transação. No caso de operações do tipo **authorize** e **capture**, quando a transação passa como **psp**, são geradas duas linhas, uma com valor **pagarme** e outra com o nome da adquirente. Se for **gateway**, tem-se apenas uma linha de autorização e captura com o nome da adquirente;
- **transaction_operation_created_at** (*timestampz*): Horário de criação da operação;

Uma característica do produto pagar.me é a retentativa entre adquirentes. Caso, a autorização de uma transação seja negada por uma adquirente, tenta-se em outra adquirente, o que pode aumentar a conversão. Todas as tentativas são registradas nesta tabela.

Obs1: A única maneira de se encontrar o horário de captura ou estorno de transações que passaram como **gateway** é por meio do campo **transaction_operation_created_at** da tabela "TransactionOperations" para os types **capture** e **refund**, respectivamente.

Obs2: Adote que a transação está capturada quando as seguintes condições são satisfeitas:

- **status=success**, **type=capture** e **acquirer_name=pagarme** para transações psp

- **status**=success, **type**=capture e **acquirer_name**=acquirer_x, onde acquirer_x representa o nome de uma possível adquirente, para transações gateway.