FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

RM99565 – Erick Nathan Capito Pereira

RM550841 – Lucas Araujo Oliveira Silva

RM99409 – Michael José Bernardi da Silva

RM99577 – Guilherme Dias Gomes



São Paulo, SP

2024

Sumário

[Guia de arquivos da entrega 3](#_Toc176812255)

[Funções 4](#_Toc176812256)

[1. fn\_convert\_to\_json 4](#_Toc176812257)

[2. fn\_calculate\_event\_revenue 5](#_Toc176812258)

[Procedures 6](#_Toc176812259)

[1. proc\_column\_value\_comparison 6](#_Toc176812260)

[2. proc\_get\_user\_events 7](#_Toc176812261)

[Trigger 7](#_Toc176812262)

[1. tbiud\_event 7](#_Toc176812263)

# Guia de arquivos da entrega

|  |  |
| --- | --- |
| NOME DO ARQUIVO/PASTA | DESCRIÇÃO |
| scripts/geniusxp\_dml.sql | Contém scripts para realizar a inserção de dados nas tabelas. |
| scripts/geniusxp\_ddl.sql | Contém scripts para a criação das tabelas e sequences. |
| scripts/geniusxp\_functions.sql | Contém scripts das funções criadas. |
| scripts/geniusxp\_procedures.sql | Contém scripts das procedures criadas. |
| scripts/geniusxp\_procedures.sql | Contém script da trigger de auditoria. |
| scripts/geniusxp\_types.sql | Contém scripts dos tipos criados para utilização nos demais objetos. |
| autochef\_limpeza\_dados.sql | Contém scripts para a remoção das funções, procedures e tabelas. |
| modelagens | Contém os modelos lógico e físico do banco de dados. |
| geniusxp\_apresentacao\_da\_ideia.pdf | Contém a apresentação da ideia, tais como as funcionalidades, público alvo e descrição do problema |
| prints | Contém os prints de execução dos scripts. |

# Funções

## fn\_convert\_to\_json

**Descrição:** A função fn\_convert\_to\_json tem como objetivo converter uma coleção de registros (p\_user\_events) em um formato JSON estruturado. Esta função é útil para transformar dados de eventos de um usuário em um formato facilmente manipulável e compartilhável, como JSON.

**Parâmetros:**

* p\_user\_events (get\_user\_events\_table\_type): Coleção de registros que contém informações de eventos, como nome do evento, tipo de evento, preço total, número do ticket e data de uso.

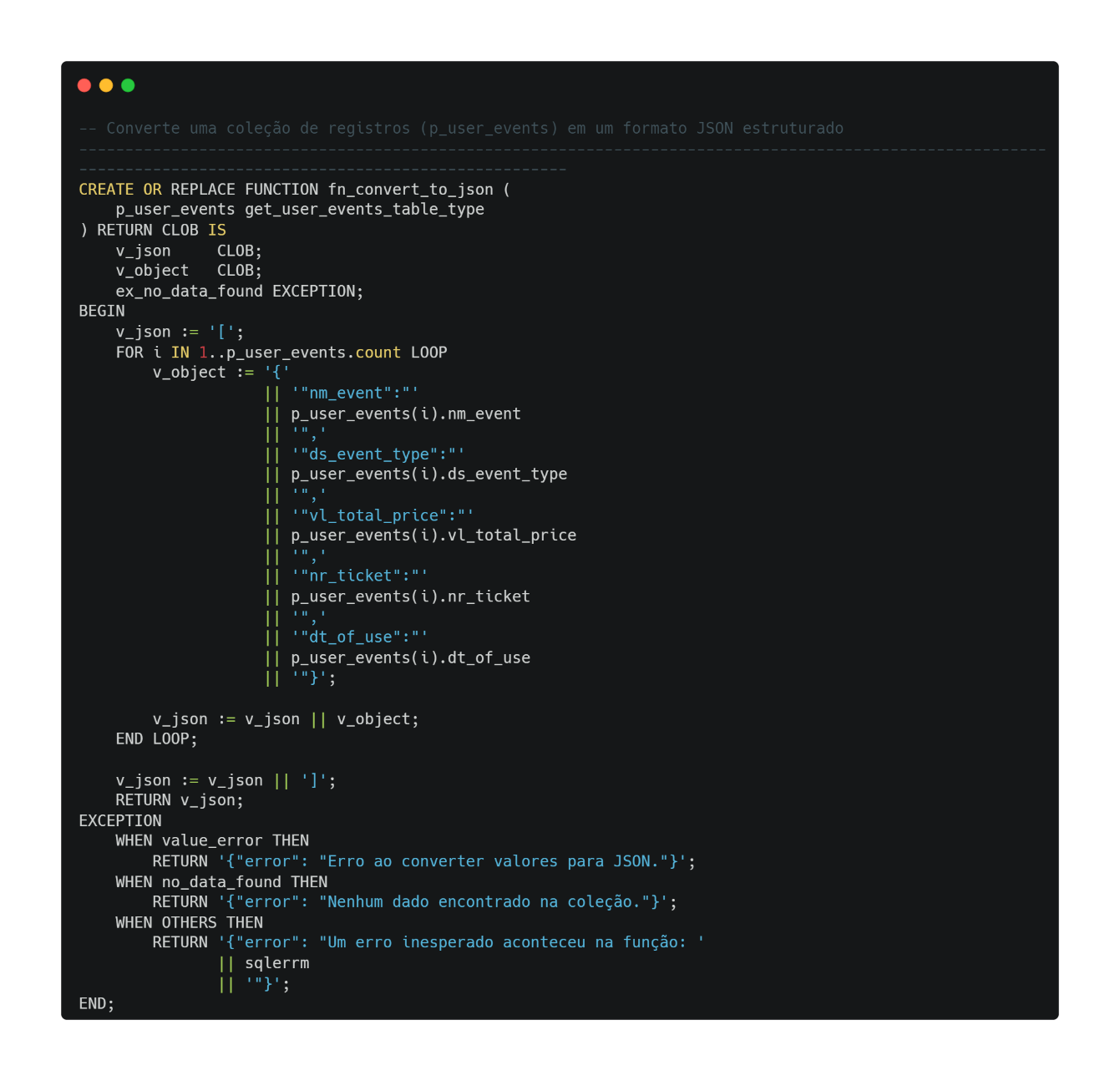
**Exceções tratadas:**

* value\_error: Captura erros que ocorrem ao converter valores para o formato JSON.
* ex\_no\_data\_found: Retorna uma mensagem de erro JSON quando a coleção de registros está vazia.
* OTHERS: Captura qualquer outro erro inesperado, retornando uma mensagem de erro genérica com detalhes do erro.

**Retorno:**

* CLOB: A função retorna uma string em formato JSON que representa a coleção de registros fornecida.

**Código da função:**



## fn\_calculate\_event\_revenue

**Descrição:** A função fn\_calculate\_event\_revenue calcula a receita total gerada por um evento específico com base no ID do evento fornecido. A receita é calculada somando os valores de pagamentos confirmados (status 'Aprovado' ou 'Concluído') para os tickets associados ao evento.

**Parâmetros:**

* p\_event\_id (T\_GXP\_EVENT.id\_event%TYPE): Identificador do evento

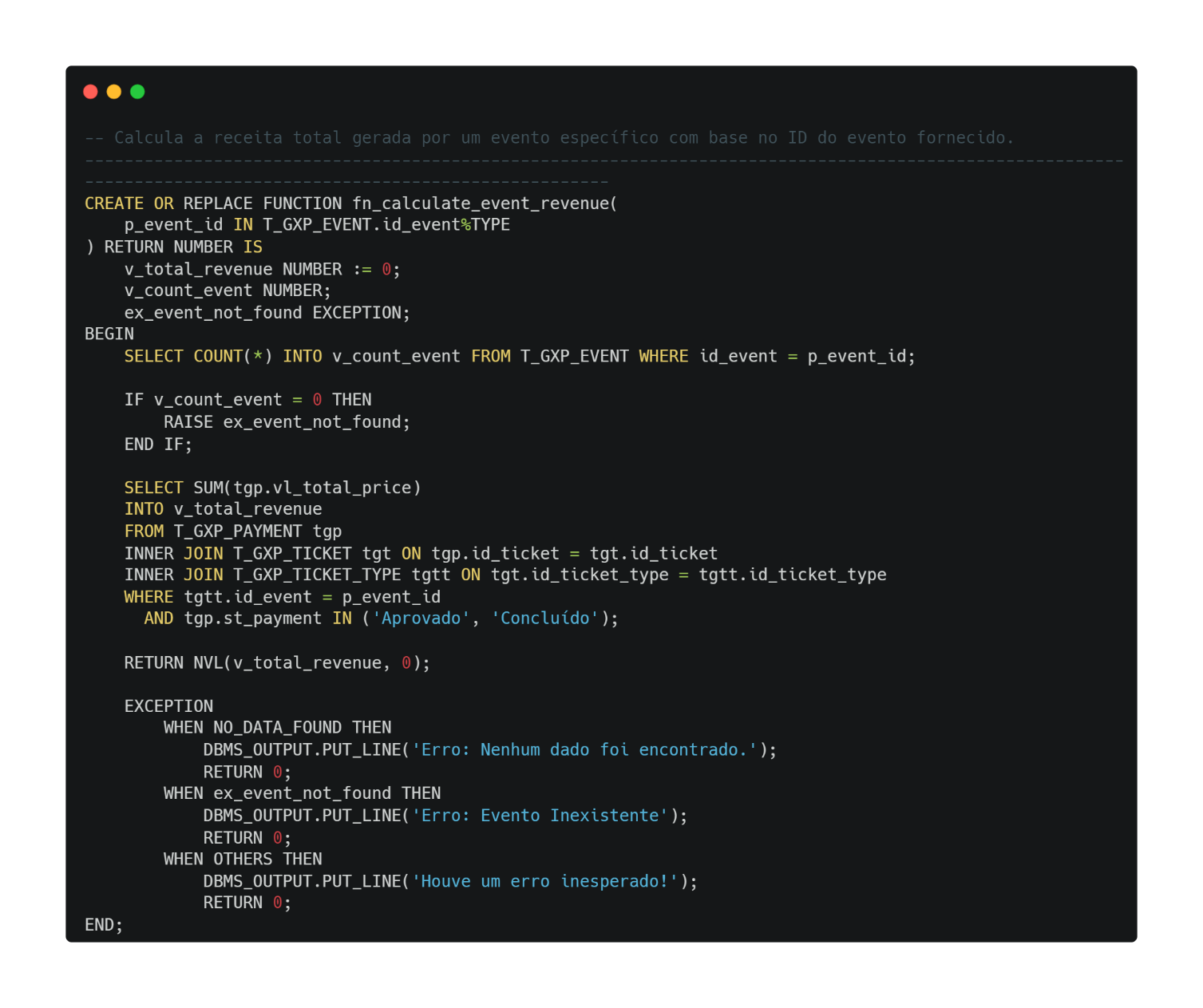
**Exceções Tratadas:**

* NO\_DATA\_FOUND: Captura a situação em que não há dados encontrados durante a soma dos valores de pagamento.
* ex\_event\_not\_found: Captura a situação onde o evento com o ID fornecido não existe
* OTHERS: Captura qualquer outro erro inesperado, retornando uma mensagem de erro genérica com detalhes do erro.

**Retorno:**

* NUMBER: A função retorna o total da receita gerada pelo evento.

**Código da função:**

****

# Procedures

## proc\_column\_value\_comparison

**Descrição:** A procedure proc\_column\_value\_comparison tem como objetivo realizar a comparação entre valores de eventos consecutivos na tabela t\_gxp\_event. Para cada evento, a procedure compara o nome do evento atual com o nome do evento anterior e do próximo, utilizando as funções analíticas LAG e LEAD. O resultado da comparação é exibido no console usando DBMS\_OUTPUT.PUT\_LINE.

**Parâmetros:**

* Nenhum parâmetro de entrada ou saída é necessário.

**Exceções Tratadas:**

* ex\_insufficient\_data: Lançada quando a consulta retorna menos de 5 registros.
* ex\_query\_failure: Lançada caso ocorra um erro na consulta SQL ou na coleta dos dados.
* OTHERS: Captura qualquer outra exceção inesperada

**Retorno:**

* A procedure imprime no console as informações sobre o evento atual, o evento anterior e o próximo evento para cada registro processado.

## proc\_get\_user\_events

**Descrição:** A procedure proc\_generate\_user\_events tem como objetivo gerar um JSON contendo os eventos associados a um usuário específico, com base no seu ID. A procedure faz uso de uma função personalizada (fn\_convert\_to\_json) para converter os dados retornados pelo cursor em formato JSON.

**Parâmetros:**

* p\_user\_id (IN t\_gxp\_user.id\_user%TYPE): ID do usuário para o qual os eventos serão buscados.
* p\_json\_result (OUT CLOB): Resultado em formato JSON retornado como saída.

**Exceções Tratadas:**

* ex\_no\_data\_found: Lançada quando não são encontrados eventos associados ao usuário.
* ex\_non\_existent\_user: Lançada quando o ID do usuário fornecido não existe na base de dados.
* value\_error: Lançada em caso de erros de valor durante o processamento.
* OTHERS: Captura qualquer outra exceção inesperada.

**Retorno:**

* A procedure retorna um valor em formato JSON através do parâmetro p\_json\_result.

# Trigger

## tbiud\_event

**Descrição:** A trigger tbiud\_event é uma trigger de auditoria que registra operações de inserção, atualização e exclusão na tabela T\_GXP\_EVENT. Sempre que um registro é inserido, atualizado ou excluído na tabela T\_GXP\_EVENT, a trigger insere um novo registro na tabela T\_GXP\_AUDIT, mantendo um histórico detalhado das alterações realizadas.

**Evento:**

* BEFORE DELETE OR INSERT OR UPDATE: trigger é acionada antes das operações de exclusão, inserção e atualização na tabela T\_GXP\_EVENT.

**Operações Realizadas:**

* Define o tipo de operação realizada (INSERT, UPDATE ou DELETE) com base na ação que disparou a trigger.
* Insere um registro na tabela T\_GXP\_AUDIT.

**Exceções Tratadas:**

* OTHERS: Captura qualquer erro inesperado durante a execução da trigger e imprime uma mensagem de erro usando DBMS\_OUTPUT.PUT\_LINE.