

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

RM98110 - André Rohregger Machado

RM99565 - Erick Nathan Capito Pereira

RM550841 – Lucas Araujo Oliveira Silva

RM551886 - Victor Luca do Nascimento Queiroz

RM99455 - Vinícius Martins Torres Abdala



Sumário

Guia de arquivos da entrega.....	3
Funções para validar entrada de dados	4
1. f_validate_tb_user	4
2. f_validate_tb_order	4
3. f_validate_tb_order_item	5
4. f_validate_tb_product	5
5. f_validate_tb_restaurant	5
Procedures de INSERT, UPDATE e DELETE	6
1. add_user	6
2. remove_user	7
3. update_user	7
4. add_order	8
5. remove_order	8
6. update_order	9
7. add_order_item	9
8. remove_order_item	10
9. update_order_item	11
10. add_product	11
11. remove_product	12
12. update_product	12
13. add_restaurant	13
14. remove_restaurant	14
15. update_restaurant	14
Procedures para a geração de relatórios	15
1. most_requested_product_by_customer	15
2. not_delivered_orders	16
Implementação da extensão cx_oracle do Python	16

Guia de arquivos da entrega

NOME DO ARQUIVO/PASTA	DESCRIÇÃO
autochef_dml.sql	Contém scripts para realizar a inserção de dados nas tabelas.
autochef_ddl.sql	Contém scripts para a criação das tabelas e sequences.
autochef_functions.sql	Contém scripts das funções de validação de entrada de dados.
autochef_procedures.sql	Contém scripts das procedures de insert, delete e update das tabelas, além de procedures para geração de relatórios.
autochef_prints	Contém prints demonstrando a criação e execução dos scripts.
autochef_python_script	Contém projeto Python com a utilização da extensão cx_oracle para a chamada das procedures.
autochef_limpeza_dados.sql	Contém scripts para a remoção das funções, procedures e tabelas.
autochef_modelos	Contém os modelos lógico e físico do banco de dados.
autochef_apresentacao_da_ideia.pdf	Contém a apresentação da ideia, tais como as funcionalidades, público alvo e descrição do problema

Funções para validar entrada de dados

1. f_validate_tb_user

Descrição: A função `f_validate_tb_user` realiza uma série de validações sobre os parâmetros fornecidos para garantir que eles atendem aos critérios especificados. A função retorna um valor `BOOLEANO` indicando se todas as validações foram bem-sucedidas (`TRUE`) ou se alguma delas falhou (`FALSE`).

Parâmetros:

- `p_ds_email` (`tb_user.ds_email%TYPE`): O endereço de e-mail a ser validado.
- `p_ds_password` (`tb_user.ds_password%TYPE`): A senha a ser validada.
- `p_dt_registration` (`tb_user.dt_registration%TYPE`): A data de registro a ser validada.
- `p_nr_cpf` (`tb_user.nr_cpf%TYPE`): O CPF a ser validado.
- `p_dt_birth` (`tb_user.dt_birth%TYPE`): A data de nascimento a ser validada.

Validações Realizadas:

- O e-mail deve corresponder ao padrão regex especificado: `^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'`.
- A senha deve ter pelo menos 8 caracteres.
- A data de registro (`p_dt_registration`) não pode ser uma data futura.
- A função `f_cpf_validation` é chamada para validar o CPF.
- A data de nascimento (`p_dt_birth`) não pode ser uma data futura ou a data atual.

2. f_validate_tb_order

Descrição: A função `f_validate_tb_order` realiza validações para garantir que um pedido (order) é válido com base em determinados critérios, como a existência do usuário e do restaurante e o status do pedido. A função retorna um valor `BOOLEANO` indicando se todas as validações foram bem-sucedidas (`TRUE`) ou se alguma delas falhou (`FALSE`).

Parâmetros:

- `p_id_user` (`tb_order.id_user%TYPE`): O ID do usuário associado ao pedido.
- `p_id_restaurant` (`tb_order.id_restaurant%TYPE`): O ID do restaurante associado ao pedido.
- `p_st_order` (`tb_order.st_order%TYPE`): O status do pedido.

Validações Realizadas:

- Verifica se o usuário com o `p_id_user` fornecido existe na tabela `tb_user`. A senha deve ter pelo menos 8 caracteres.
- Verifica se o restaurante com o `p_id_restaurant` fornecido existe na tabela `tb_restaurant`.

- Verifica se o status do pedido (p_st_order) está entre os valores permitidos: 'ACCEPTED', 'DELIVERED', 'PENDING', 'REJECTED'. A data de nascimento (p_dt_birth)

3. f_validate_tb_order_item

Descrição: A função f_validate_tb_order_item realiza validações para garantir que um item de pedido é válido com base na existência do pedido e do produto associados. A função retorna um valor BOOLEANO indicando se todas as validações foram bem-sucedidas (TRUE) ou se alguma delas falhou (FALSE).

Parâmetros:

- p_id_order (tb_order_item.id_order%TYPE): O ID do pedido associado ao item de pedido.
- p_id_product (tb_order_item.id_product%TYPE): O ID do produto associado ao item de pedido.

Validações Realizadas:

- Verifica se o pedido com o p_id_order fornecido existe na tabela tb_order.
- Verifica se o produto com o p_id_product fornecido existe na tabela tb_product.

4. f_validate_tb_product

Descrição: A função f_validate_tb_product realiza uma validação para garantir que um produto é associado a um restaurante existente. A função retorna um valor BOOLEANO indicando se a validação foi bem-sucedida (TRUE) ou se falhou (FALSE).

Parâmetros:

- p_id_restaurant (tb_product.id_restaurant%TYPE): O ID do restaurante associado ao produto.

Validações Realizadas:

- Verifica se o restaurante com o p_id_restaurant fornecido existe na tabela tb_restaurant.

5. f_validate_tb_restaurant

Descrição: A função f_validate_tb_restaurant realiza uma série de validações sobre os parâmetros fornecidos para garantir que eles atendem aos critérios especificados.

A função retorna um valor **BOOLEANO** indicando se todas as validações foram bem-sucedidas (**TRUE**) ou se alguma delas falhou (**FALSE**).

Parâmetros:

- **p_ds_password** (tb_restaurant.ds_password%TYPE): A senha do restaurante a ser validada.
- **p_ds_email** (tb_restaurant.ds_email%TYPE): O endereço de e-mail do restaurante a ser validado.
- **p_nr_cnpj** (tb_restaurant.nr_cnpj%TYPE): O CNPJ do restaurante a ser validado.

Validações Realizadas:

- O e-mail deve corresponder ao padrão regex especificado: `'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'`.
- A senha deve ter pelo menos 8 caracteres.
- A função `f_cnpj_validation` é chamada para validar o CNPJ.

Procedures de **INSERT, UPDATE e DELETE**

1. **add_user**

Descrição: A procedure `add_user` insere um novo usuário na tabela `tb_user` após validar os dados fornecidos. Utiliza a função `f_validate_tb_user` para garantir que os dados atendem aos critérios especificados antes de realizar a inserção. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída `p_result`.

Parâmetros:

- **p_nm_user** (IN tb_user.nm_user%TYPE): O nome do usuário.
- **p_ds_email** (IN tb_user.ds_email%TYPE): O endereço de e-mail do usuário.
- **p_ds_password** (IN tb_user.ds_password%TYPE): A senha do usuário.
- **p_dt_registration** (IN tb_user.dt_registration%TYPE): A data de registro do usuário.
- **p_dt_birth** (IN tb_user.dt_birth%TYPE): A data de nascimento do usuário.
- **p_nr_cpf** (IN tb_user.nr_cpf%TYPE): O CPF do usuário.
- **p_result** (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função `f_validate_tb_user` para verificar se os dados fornecidos são válidos.
- Se a função `f_validate_tb_user` retornar **FALSE**, a procedure define `p_result` como 0 e não insere o usuário.

Operações Realizadas:

- Insere um novo registro na tabela `tb_user` com os valores fornecidos.

- Utiliza a sequência seq_tb_user.NEXTVAL para gerar o ID do usuário.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

2. remove_user

Descrição: A procedure remove_user remove um usuário da tabela tb_user com base no ID do usuário fornecido. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_user (IN tb_user.id_user%TYPE): O ID do usuário a ser removido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Operações Realizadas:

- Remove o registro da tabela tb_user onde id_user é igual ao valor fornecido em p_id_user.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1 para indicar sucesso.

3. update_user

Descrição: A procedure update_user atualiza os dados de um usuário na tabela tb_user com base no ID do usuário fornecido. Antes de realizar a atualização, valida os dados utilizando a função f_validate_tb_user. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_user (IN tb_user.id_user%TYPE): O ID do usuário a ser atualizado.
- p_nm_user (IN tb_user.nm_user%TYPE): O nome do usuário.
- p_ds_email (IN tb_user.ds_email%TYPE): O endereço de e-mail do usuário.
- p_ds_password (IN tb_user.ds_password%TYPE): A senha do usuário.
- p_dt_registration (IN tb_user.dt_registration%TYPE): A data de registro do usuário.
- p_dt_birth (IN tb_user.dt_birth%TYPE): A data de nascimento do usuário.
- p_nr_cpf (IN tb_user.nr_cpf%TYPE): O CPF do usuário.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_user para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_user retornar FALSE, a procedure define p_result como 0 e não atualiza o usuário.

Operações Realizadas:

- Atualiza o registro na tabela tb_user onde id_user é igual ao valor fornecido em p_id_user com os novos valores fornecidos.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

4. add_order

Descrição: A procedure add_order insere um novo pedido na tabela tb_order após validar os dados fornecidos. Utiliza a função f_validate_tb_order para garantir que os dados atendem aos critérios especificados antes de realizar a inserção. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_user (IN tb_order.id_user%TYPE): O ID do usuário que fez o pedido.
- p_id_restaurant (IN tb_order.id_restaurant%TYPE): O ID do restaurante onde o pedido foi feito.
- p_st_order (IN tb_order.st_order%TYPE): O status do pedido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_order para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_order retornar FALSE, a procedure define p_result como 0 e não insere o pedido.

Operações Realizadas:

- Insere um novo registro na tabela tb_order com os valores fornecidos.
- Utiliza a sequência seq_tb_order.NEXTVAL para gerar o ID do pedido.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

5. remove_order

Descrição: A procedure remove_order remove um pedido da tabela tb_order com base no ID do pedido fornecido. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_order (IN tb_order.id_order%TYPE): O ID do pedido a ser removido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Operações Realizadas:

- Remove o registro da tabela tb_order onde id_order é igual ao valor fornecido em p_id_order.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1 para indicar sucesso.

6. update_order

Descrição: A procedure update_order atualiza os dados de um pedido na tabela tb_order com base no ID do pedido fornecido. Antes de realizar a atualização, valida os dados utilizando a função f_validate_tb_order. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_order (IN tb_order.id_order%TYPE): O ID do pedido a ser atualizado.
- p_id_user (IN tb_order.id_user%TYPE): O ID do usuário associado ao pedido.
- p_id_restaurant (IN tb_order.id_restaurant%TYPE): O ID do restaurante associado ao pedido.
- p_st_order (IN tb_order.st_order%TYPE): O status do pedido.
- p_ds_feedback (IN tb_order.ds_feedback%TYPE): O feedback do pedido.
- p_nr_rate (IN tb_order.nr_rate%TYPE): A avaliação do pedido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_order para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_order retornar FALSE, a procedure define p_result como 0 e não atualiza o pedido.

Operações Realizadas:

- Atualiza o registro na tabela tb_order onde id_order é igual ao valor fornecido em p_id_order com os novos valores fornecidos.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

7. add_order_item

Descrição: A procedure add_order_item insere um novo item de pedido na tabela tb_order_item após validar os dados fornecidos. Utiliza a função f_validate_tb_order_item para garantir que os dados atendem aos critérios especificados antes de realizar a inserção. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_order (IN tb_order_item.id_order%TYPE): O ID do pedido ao qual o item pertence.
- p_id_product (IN tb_order_item.id_product%TYPE): O ID do produto associado ao item.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_order_item para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_order_item retornar FALSE, a procedure define p_result como 0 e não insere o item de pedido.

Operações Realizadas:

- Insere um novo registro na tabela tb_order_item com os valores fornecidos.
- Utiliza a sequência seq_tb_order_item.NEXTVAL para gerar o ID do item de pedido.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

8. remove_order_item

Descrição: A procedure remove_order_item remove um item de pedido da tabela tb_order_item com base no ID do item fornecido. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_item (IN tb_order_item.id_item%TYPE): O ID do item de pedido a ser removido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Operações Realizadas:

- Remove o registro da tabela tb_order_item onde id_item é igual ao valor fornecido em p_id_item.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1 para indicar sucesso.

9. update_order_item

Descrição: A procedure update_order_item atualiza os dados de um item de pedido na tabela tb_order_item com base no ID do item fornecido. Antes de realizar a atualização, valida os dados utilizando a função f_validate_tb_order_item. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_item (IN tb_order_item.id_item%TYPE): O ID do item de pedido a ser atualizado.
- p_id_order (IN tb_order_item.id_order%TYPE): O ID do pedido ao qual o item pertence.
- p_id_product (IN tb_order_item.id_product%TYPE): O ID do produto associado ao item.
- p_ds_order_chance (IN tb_order_item.ds_order_chance%TYPE): A descrição da chance do pedido.
- p_ds_amount (IN tb_order_item.ds_amount%TYPE): A quantidade do pedido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_order_item para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_order_item retornar FALSE, a procedure define p_result como 0 e não atualiza o item de pedido.

Operações Realizadas:

- Atualiza o registro na tabela tb_order_item onde id_item é igual ao valor fornecido em p_id_item com os novos valores fornecidos.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

10. add_product

Descrição: A procedure add_product adiciona um novo produto na tabela tb_product após validar os dados fornecidos. Utiliza a função f_validate_tb_product para garantir que os dados atendem aos critérios especificados antes de realizar a inserção. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_restaurant (IN tb_product.id_restaurant%TYPE): O ID do restaurante ao qual o produto pertence.
- p_ds_image_url (IN tb_product.ds_image_url%TYPE): O URL da imagem do produto.
- p_nm_product (IN tb_product.nm_product%TYPE): O nome do produto.

- p_nr_price (IN tb_product.nr_price%TYPE): O preço do produto.
- p_ds_category (IN tb_product.ds_category%TYPE): A categoria do produto.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_product para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_product retornar FALSE, a procedure define p_result como 0 e não insere o produto.

Operações Realizadas:

- Insere um novo registro na tabela tb_product com os valores fornecidos.
- Utiliza a sequência seq_tb_product.NEXTVAL para gerar o ID do produto.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

11. remove_product

Descrição: A procedure remove_product remove um produto da tabela tb_product com base no ID do produto fornecido. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_product (IN tb_product.id_product%TYPE): O ID do produto a ser removido.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Operações Realizadas:

- Remove o registro da tabela tb_product onde id_product é igual ao valor fornecido em p_id_product.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1 para indicar sucesso.

12. update_product

Descrição: A procedure update_product atualiza os dados de um produto na tabela tb_product com base no ID do produto fornecido. Antes de realizar a atualização, valida os dados utilizando a função f_validate_tb_product. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_id_product (IN tb_product.id_product%TYPE): O ID do produto a ser atualizado.
- p_id_restaurant (IN tb_product.id_restaurant%TYPE): O ID do restaurante ao qual o produto pertence.
- p_ds_image_url (IN tb_product.ds_image_url%TYPE): O URL da imagem do produto.
- p_nm_product (IN tb_product.nm_product%TYPE): O nome do produto.
- p_nr_price (IN tb_product.nr_price%TYPE): O preço do produto.
- p_ds_category (IN tb_product.ds_category%TYPE): A categoria do produto.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_product para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_product retornar FALSE, a procedure define p_result como 0 e não atualiza o produto.

Operações Realizadas:

- Atualiza o registro na tabela tb_product onde id_product é igual ao valor fornecido em p_id_product com os novos valores fornecidos.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

13. add_restaurant

Descrição: A procedure add_restaurant adiciona um novo restaurante na tabela tb_restaurant após validar os dados fornecidos. Utiliza a função f_validate_tb_restaurant para garantir que os dados atendem aos critérios especificados antes de realizar a inserção. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída p_result.

Parâmetros:

- p_ds_password (IN tb_restaurant.ds_password%TYPE): A senha do restaurante.
- p_ds_email (IN tb_restaurant.ds_email%TYPE): O email do restaurante.
- p_ds_image_url (IN tb_restaurant.ds_image_url%TYPE): O URL da imagem do restaurante.
- p_nr_cnpj (IN tb_restaurant.nr_cnpj%TYPE): O CNPJ do restaurante.
- p_nm_restaurant (IN tb_restaurant.nm_restaurant%TYPE): O nome do restaurante.
- p_nr_longitude (IN tb_restaurant.nr_longitude%TYPE): A longitude do restaurante.
- p_nr_latitude (IN tb_restaurant.nr_latitude%TYPE): A latitude do restaurante.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_restaurant para verificar se os dados fornecidos são válidos.

- Se a função `f_validate_tb_restaurant` retornar `FALSE`, a procedure define `p_result` como 0 e não insere o restaurante.

Operações Realizadas:

- Insere um novo registro na tabela `tb_restaurant` com os valores fornecidos.
- Utiliza a sequência `seq_tb_restaurant.NEXTVAL` para gerar o ID do restaurante.
- Realiza um `COMMIT` para salvar a transação.
- Define `p_result` como 1.

14. remove_restaurant

Descrição: A procedure `remove_restaurant` remove um restaurante da tabela `tb_restaurant` com base no ID do restaurante fornecido. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída `p_result`.

Parâmetros:

- `p_id_restaurant` (IN `tb_restaurant.id_restaurant%TYPE`): O ID do restaurante a ser removido.
- `p_result` (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Operações Realizadas:

- Remove o registro da tabela `tb_restaurant` onde `id_restaurant` é igual ao valor fornecido em `p_id_restaurant`.
- Realiza um `COMMIT` para salvar a transação.
- Define `p_result` como 1 para indicar sucesso.

15. update_restaurant

Descrição: A procedure `update_restaurant` atualiza os dados de um restaurante na tabela `tb_restaurant` com base no ID do restaurante fornecido. Antes de realizar a atualização, valida os dados utilizando a função `f_validate_tb_restaurant`. A procedure retorna um indicador de sucesso ou falha através do parâmetro de saída `p_result`.

Parâmetros:

- `p_id_restaurant` (IN `tb_restaurant.id_restaurant%TYPE`): O ID do restaurante a ser atualizado.
- `p_ds_password` (IN `tb_restaurant.ds_password%TYPE`): A senha do restaurante.
- `p_ds_email` (IN `tb_restaurant.ds_email%TYPE`): O email do restaurante.

- p_ds_image_url (IN tb_restaurant.ds_image_url%TYPE): O URL da imagem do restaurante.
- p_nr_cnpj (IN tb_restaurant.nr_cnpj%TYPE): O CNPJ do restaurante.
- p_nm_restaurant (IN tb_restaurant.nm_restaurant%TYPE): O nome do restaurante.
- p_nr_longitude (IN tb_restaurant.nr_longitude%TYPE): A longitude do restaurante.
- p_nr_latitude (IN tb_restaurant.nr_latitude%TYPE): A latitude do restaurante.
- p_result (OUT INT): Resultado da operação (1 para sucesso, 0 para falha).

Validações Realizadas:

- Utiliza a função f_validate_tb_restaurant para verificar se os dados fornecidos são válidos.
- Se a função f_validate_tb_restaurant retornar FALSE, a procedure define p_result como 0 e não atualiza o restaurante.

Operações Realizadas:

- Atualiza o registro na tabela tb_restaurant onde id_restaurant é igual ao valor fornecido em p_id_restaurant com os novos valores fornecidos.
- Realiza um COMMIT para salvar a transação.
- Define p_result como 1.

Procedures para a geração de relatórios

1. most_requested_product_by_customer

Descrição: A procedure most_requested_product_by_customer retorna o produto mais pedido por um cliente específico na tabela tb_product. O produto mais solicitado é determinado contando o número de vezes que foi incluído em pedidos feitos pelo cliente e selecionando aquele com o maior número de pedidos. O nome do produto mais pedido é retornado através do parâmetro de saída p_nm_product.

Parâmetros:

- p_id_user (IN tb_user.id_user%TYPE): O ID do usuário (cliente) para o qual o produto mais pedido será determinado.
- p_nm_product (OUT tb_product.nm_product%TYPE): O nome do produto mais pedido será retornado através deste parâmetro.

Operações Realizadas:

- Define um cursor cur_product_by_customer que seleciona o produto mais pedido para o cliente fornecido (p_id_user). O produto mais pedido é determinado contando o número de vezes que foi incluído em pedidos feitos pelo cliente e selecionando aquele com o maior número de pedidos.

- O resultado é ordenado em ordem decrescente pelo número de itens pedidos.
- Utiliza ROWNUM para limitar o resultado a apenas o produto mais pedido.
- Recupera o resultado do cursor e armazena o ID do produto, o número total de itens pedidos e o nome do produto mais pedido nas variáveis apropriadas.
- Atribui o nome do produto mais pedido ao parâmetro de saída p_nm_product.

2. not_delivered_orders

Descrição: A procedure not_delivered_orders lista todos os pedidos de um restaurante específico que ainda não foram entregues. A regra de negócio subjacente é garantir que todos os pedidos de um restaurante sejam entregues dentro do prazo, minimizando o tempo de espera dos clientes e melhorando a eficiência operacional.

Parâmetros:

- p_id_restaurant (IN tb_restaurant.id_restaurant%TYPE): O ID do restaurante para o qual os pedidos não entregues serão listados.

Operações Realizadas:

- Define um cursor cur_not_delivered_orders que seleciona todos os pedidos não entregues para o restaurante fornecido (p_id_restaurant).
- Os pedidos são selecionados da tabela tb_order com base na condição em que o status do pedido não seja 'DELIVERED' e o ID do restaurante corresponda ao fornecido.
- Os resultados são agrupados por ID do pedido e ordenados pelo número total de itens nos pedidos.
- Recupera o resultado do cursor e armazena o ID do pedido, o número total de itens no pedido e o preço total do pedido nas variáveis apropriadas.
- Exibe na saída padrão os detalhes de cada pedido não entregue, incluindo o ID do pedido, o total de itens e o preço total.

Implementação da extensão cx_oracle do Python

Foi construído um programa Python que interage com o banco de dados Oracle para inserir dados em tabelas específicas do Autochef. Para isso, é exibido um menu de opções para o usuário escolher a operação desejada:

- Inserir dados na tabela TB_USER.
- Inserir dados na tabela TB_ORDER.
- Inserir dados na tabela TB_ORDER_ITEM.

- Inserir dados na tabela TB_PRODUCT.
- Inserir dados na tabela TB_RESTAURANT.
- Sair do programa.

Dependendo da opção escolhida pelo usuário, o código solicita os dados necessários e chama a procedure apropriada no banco de dados para inserir os dados.

OBS: Para utilizar o programa e acessar o Banco de Dados Oracle, defina o **user** e a **password** no arquivo **env.json**.