Assignment 02 - Chamar um procedimento armazenado com parâmetros IN OUT

Vamos supor que o procedimento armazenado product\_report exista com a seguinte definição:

product\_report(productId **IN** products.product\_id%type,  
        categoryId **IN OUT** products.category\_id%type,  
        productName **OUT** products.product\_name%type)

Este procedimento recebe três parâmetros;**IN,** **IN OUT** e **OUT**.

Este procedimento procura productId e categoryId na tabela de produtos. Se o produto com este productId e categoryId existir, ele armazenará os valores das colunas category\_id e product-name nos parâmetros correspondentes.

Se o produto com productId e categoryId fornecidos não existir, o procedimento armazena zero e 'Not Found' aos parâmetros correspondentes:

categoryId := 0;  
productName := 'Not Found';

O exemplo a seguir mostra como chamar product\_report em uma função C++.

void **displayProduct**(Connection\* conn, int productId, int categoryId) {

    string productName;  
    Statement\* stmt = nullptr;  
    stmt = conn->createStatement("BEGIN product\_category(:1, :**2,** :3); END;");  
    stmt->setInt(1, productId);  
  **stmt->setInt(2, categoryId);**  
    stmt->registerOutParam(3, Type::OCCISTRING, sizeof(productName));  
    stmt->execute();  
    **categoryId = stmt->getInt(2);**  
    productName = stmt->getString(3);  
}