**PROCEDURA – REALIZUJ ZAMÓWIENIE**

1. SPRAWDZA CZY WIELKOŚCI KOLEKCJI ODPOWIADAJACYCH ZA ILOSC ZABAWEK ORAZ LICZBE ILOSCI ZAMÓWIONYCH ZABAWEKSĄ RÓWNE
2. JEŚLI NIE – STOSOWNY KOMUNIKAT, STOP
3. JEŚLI TAK – KONTYNUUJ
4. WSTAWIA NOWY WIERSZ DO TABELI ZAMOWIENIA
5. SPRAWDZA CZY JEST DOSTEPNA WYSTARCZAJACA ILOSC ZABAWEK NA MAGAZYNIE
6. JEŚLI NIE – STOSOWNY KOMUNIKAT, STOP
7. JEŚLI TAK - KONTYNUUJ
8. ZLICZA ILOŚĆ ZABAWEK ZAKUPIONYCH PRZEZ KLIENTA
9. SPRAWDZA CZY ILOŚĆ TA JEST WIĘKSZA NIŻ 100
10. Jeśli tak – zmienia klienta na stałego
11. Jeśli nie - kontynuuj
12. OBSŁUGA WYJĄTKÓW

CREATE TYPE ZabawkiCollection AS TABLE OF INTEGER;

CREATE TYPE IlosciZabawekCollection AS TABLE OF INTEGER;

CREATE OR REPLACE PROCEDURE RealizujZamowienie(

p\_id\_klienta INTEGER,

p\_data\_zamowienia DATE,

p\_id\_sposobu\_zaplaty INTEGER,

p\_produkty\_tab IN ZabawkiCollection,

p\_ilosci\_tab IN IlosciZabawekCollection

) AS

v\_id\_zamowienia INTEGER;

v\_id\_pozycji\_zamowienia INTEGER;

v\_liczba\_zamowionych\_zabawek INTEGER := 0;

BEGIN

-- Sprawdź, czy ilości i produkty mają taką samą długość

IF p\_produkty\_tab.COUNT <> p\_ilosci\_tab.COUNT THEN

DBMS\_OUTPUT.PUT\_LINE('Błąd: Kolekcje p\_produkty\_tab i p\_ilosci\_tab muszą mieć tę samą długość.');

RETURN;

END IF;

-- Znajdź pierwszy dostępny identyfikator zamówienia

SELECT COALESCE(MAX(id\_zamowienia) + 1, 1)

INTO v\_id\_zamowienia

FROM Zamowienia;

-- Znajdź pierwszy dostępny identyfikator pozycji zamówienia

SELECT COALESCE(MAX(id\_pozycji\_zamowienia) + 1, 1)

INTO v\_id\_pozycji\_zamowienia

FROM Pozycje\_Zamowienia;

-- Rozpocznij transakcję

BEGIN

-- Wstaw dane zamówienia do tabeli Zamowienia

INSERT INTO Zamowienia (id\_zamowienia, id\_klienta, data\_zamowienia, status\_zamowienia, id\_sposobu\_zaplaty)

VALUES (v\_id\_zamowienia, p\_id\_klienta, p\_data\_zamowienia, 'W trakcie', p\_id\_sposobu\_zaplaty);

-- Wstaw produkty do tabeli Pozycje\_Zamowienia

FOR i IN 1..p\_produkty\_tab.COUNT LOOP

-- Sprawdź dostępność zabawek

DECLARE

v\_dostepna\_ilosc INTEGER;

BEGIN

SELECT dostepna\_ilosc

INTO v\_dostepna\_ilosc

FROM Zabawki

WHERE id\_zabawki = p\_produkty\_tab(i);

IF v\_dostepna\_ilosc < p\_ilosci\_tab(i) THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Błąd: Brak wystarczającej ilości zabawek o ID ' || p\_produkty\_tab(i));

END IF;

-- Odejmij zamówione ilości od dostępnych zabawek

UPDATE Zabawki

SET dostepna\_ilosc = dostepna\_ilosc - p\_ilosci\_tab(i)

WHERE id\_zabawki = p\_produkty\_tab(i);

-- Zlicz zamówione ilości

v\_liczba\_zamowionych\_zabawek := v\_liczba\_zamowionych\_zabawek + p\_ilosci\_tab(i);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Błąd: Zabawka o ID ' || p\_produkty\_tab(i) || ' nie istnieje.');

END;

-- Wstaw pozycję zamówienia

INSERT INTO Pozycje\_Zamowienia (id\_pozycji\_zamowienia, id\_zamowienia, id\_zabawki, ilosc\_sztuk)

VALUES (v\_id\_pozycji\_zamowienia, v\_id\_zamowienia, p\_produkty\_tab(i), p\_ilosci\_tab(i));

v\_id\_pozycji\_zamowienia := v\_id\_pozycji\_zamowienia + 1;

END LOOP;

-- Zakończ transakcję

COMMIT;

-- Sprawdzenie, czy liczba zamówionych zabawek przekracza 100

IF v\_liczba\_zamowionych\_zabawek > 100 THEN

-- Zmiana klienta na stałego

UPDATE Klienci

SET staly\_klient = 1

WHERE id\_klienta = p\_id\_klienta;

DBMS\_OUTPUT.PUT\_LINE('Klient o ID ' || p\_id\_klienta || ' został zmieniony na stałego klienta.');

END IF;

DBMS\_OUTPUT.PUT\_LINE('Zamówienie o ID ' || v\_id\_zamowienia || ' zostało zrealizowane.');

EXCEPTION

WHEN OTHERS THEN

-- W przypadku błędu, wykonaj rollback

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Wystąpił błąd podczas realizacji zamówienia: ' || SQLERRM);

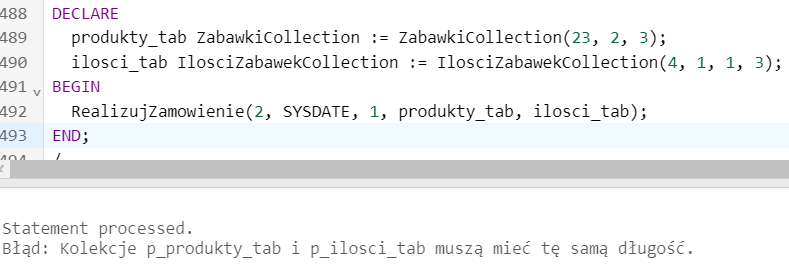
END;

END RealizujZamowienie;

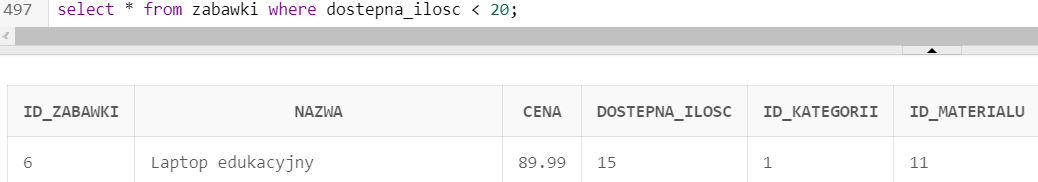
/

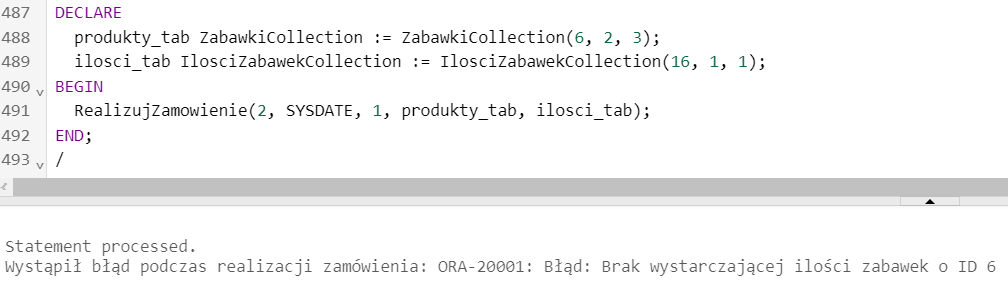
**BŁĘDY**

- BŁĄD PRZY RÓŻNEJ DŁUGOŚCI TABLIC



-BŁĄD - ZA MAŁO ZABAWEK W BAZIE

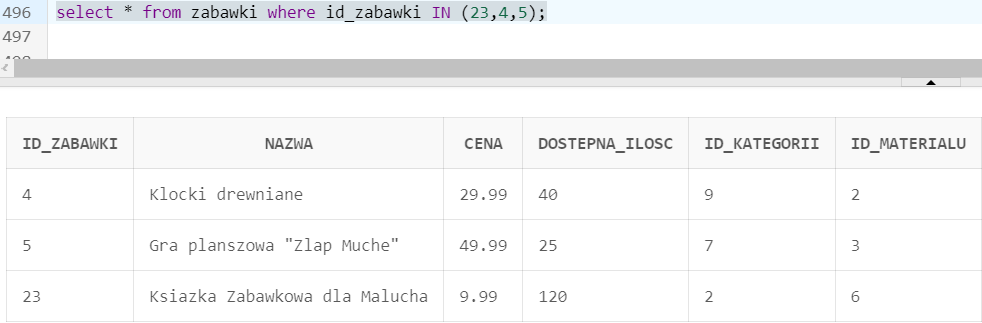




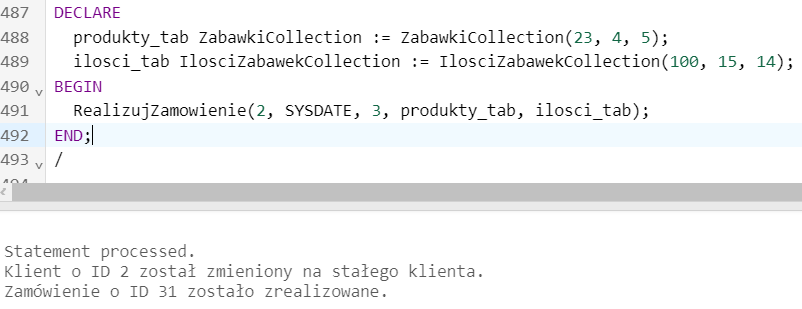
**POPRAWNE WYKONANIE**

Przed wykonaniem procedury:





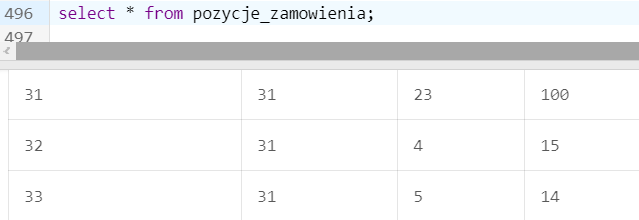
Wykonanie procedury:



Efekty działania procedury:







1. SPRAWDZA JEŚLI STATUS JEST POPRAWNY (‘W TRAKCIE’, ‘ZREALIZOWANE’, ‘ANULOWANE’).
2. JEŚLI NIE – STOSOWNY KOMUNIKAT, STOP
3. JEŚLI TAK - KONTYNUUJ
4. SPRAWDZA CZY PODANY NOWY STATUS RÓŻNI SIĘ OD OBECNEGO
5. JEŚLI NIE – STOSOWNY KOMUNIKAT, STOP
6. JEŚLI TAK - KONTYNUUJ
7. JEŚLI ZMIENIAMY Z ‘ANULOWANE’ NA (‘W TRAKCIE’, ‘ZREALIZOWANE’) TO:
8. SPRAWDZAMY CZY JEST WYSTARCZAJACA LICZBA ZABAWEK W BAZIE, ABY ZREALIZOWAC ZAMOWIENIE
9. Jeśli tak – zmniejszamy liczbę dostępnych zabawek
10. Jeśli nie – stosowny komunikat
11. SPRAWDZAMY CZY ILOSC ZAKUPIONYCH ZABAWEK PRZEZ KLIENTA PRZEKROCZYLA 100
12. Jeśli tak – zmieniamy status klienta na stałego
13. Jeśli nie – kontynuuj
14. JEŚLI ZMIENIAMY Z (‘W TRAKCIE’, ‘ZREALIZOWANE’) NA ANULOWANE TO:
15. Dodajemy liczbę dostępnych zabawek
16. SPRAWDZAMY CZY ILOSC ZAKUPIONYCH ZABAWEK PRZEZ KLIENTA NIE SPADŁA PONIZEJ 100
17. Jeśli tak – zmieniamy status klienta na niestałego
18. Jeśli nie – kontynuuj
19. OBSŁUGA WYJĄTKÓW

CREATE OR REPLACE PROCEDURE ZmienStatusZamowienia(

p\_id\_zamowienia INTEGER,

p\_nowy\_status VARCHAR2

) AS

v\_obecny\_status VARCHAR2(20);

-- Definicja kursora

CURSOR c\_zabawki IS

SELECT PZ.id\_zabawki, PZ.ilosc\_sztuk, Z.id\_klienta

FROM Pozycje\_Zamowienia PZ

JOIN Zamowienia Z ON PZ.id\_zamowienia = Z.id\_zamowienia

WHERE PZ.id\_zamowienia = p\_id\_zamowienia;

-- Zmienne do przechowywania danych z kursora

v\_zabawka\_id INTEGER;

v\_ilosc\_sztuk INTEGER;

v\_id\_klienta INTEGER;

v\_dostepna\_ilosc INTEGER;

v\_ilosc\_zamowiona\_przez\_klienta INTEGER;

BEGIN

-- Sprawdzenie, czy zamówienie istnieje

SELECT status\_zamowienia

INTO v\_obecny\_status

FROM Zamowienia

WHERE id\_zamowienia = p\_id\_zamowienia;

-- Sprawdzenie, czy podany status jest poprawny

IF p\_nowy\_status NOT IN ('W trakcie', 'Zrealizowane', 'Anulowane') THEN

DBMS\_OUTPUT.PUT\_LINE('Błąd: Niepoprawny status zamówienia.');

RETURN;

END IF;

-- Sprawdzenie, czy nowy status różni się od obecnego

IF p\_nowy\_status = v\_obecny\_status THEN

DBMS\_OUTPUT.PUT\_LINE('Błąd: Nowy status jest taki sam jak obecny.');

RETURN;

END IF;

-- Zmiana statusu zamówienia

UPDATE Zamowienia

SET status\_zamowienia = p\_nowy\_status

WHERE id\_zamowienia = p\_id\_zamowienia;

-- Otwarcie kursora

OPEN c\_zabawki;

-- Pętla FETCH dla kursora

LOOP

FETCH c\_zabawki INTO v\_zabawka\_id, v\_ilosc\_sztuk, v\_id\_klienta;

EXIT WHEN c\_zabawki%NOTFOUND;

SELECT dostepna\_ilosc

INTO v\_dostepna\_ilosc

FROM Zabawki

WHERE id\_zabawki = v\_zabawka\_id;

SELECT SUM(PZ.ilosc\_sztuk)

INTO v\_ilosc\_zamowiona\_przez\_klienta

FROM Pozycje\_Zamowienia PZ

JOIN Zamowienia Z ON PZ.id\_zamowienia = Z.id\_zamowienia

WHERE Z.id\_klienta = v\_id\_klienta;

-- Usuniecie zabawek przypadku zmiany z na "Zrealizowane" lub "W trakcie"

IF v\_obecny\_status = 'Anulowane' AND (p\_nowy\_status = 'W trakcie' OR p\_nowy\_status = 'Zrealizowane') THEN

IF(v\_dostepna\_ilosc - v\_ilosc\_sztuk < 0) THEN

DBMS\_OUTPUT.PUT\_LINE('Niewystarczajaca ilosc zabawek na przeprowadzenie takiej operacji.');

RETURN;

END IF;

UPDATE Zabawki

SET dostepna\_ilosc = dostepna\_ilosc - v\_ilosc\_sztuk

WHERE id\_zabawki = v\_zabawka\_id;

-- Sprawdzenie czy ilość kupionych zabawek przez klienta przekroczyła 100

IF ((v\_ilosc\_zamowiona\_przez\_klienta + v\_ilosc\_sztuk) > 100) THEN

-- Zmiana klienta na stałego

UPDATE Klienci

SET staly\_klient = 1

WHERE id\_klienta = v\_id\_klienta;

DBMS\_OUTPUT.PUT\_LINE('Zmieniono klienta o ID ' || v\_id\_klienta || ' na klienta stalego.');

END IF;

DBMS\_OUTPUT.PUT\_LINE('Zamówienie o ID ' || p\_id\_zamowienia || ' zostało zrealizowane, zabawki usuniete z magazynu.');

END IF;

-- Przywrocenie zabawek do magazynu w przypadku zmiany na "Anulowane"

IF (v\_obecny\_status = 'Zrealizowane' OR v\_obecny\_status = 'W trakcie') AND p\_nowy\_status = 'Anulowane' THEN

UPDATE Zabawki

SET dostepna\_ilosc = dostepna\_ilosc + v\_ilosc\_sztuk

WHERE id\_zabawki = v\_zabawka\_id;

-- Sprawdzenie czy ilość kupionych zabawek przez klienta nie spadła poniżej 100

IF ((v\_ilosc\_zamowiona\_przez\_klienta - v\_ilosc\_sztuk) < 100) THEN

-- Zmiana klienta na niestałego

UPDATE Klienci

SET staly\_klient = 0

WHERE id\_klienta = v\_id\_klienta;

DBMS\_OUTPUT.PUT\_LINE('Zmieniono klienta o ID ' || v\_id\_klienta || ' na klienta niestalego.');

END IF;

DBMS\_OUTPUT.PUT\_LINE('Zamówienie o ID ' || p\_id\_zamowienia || ' zostało anulowane, zabawki przywrocone do magazynu.');

END IF;

END LOOP;

CLOSE c\_zabawki;

DBMS\_OUTPUT.PUT\_LINE('Status zamówienia o ID ' || p\_id\_zamowienia || ' został zmieniony na: ' || p\_nowy\_status);

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Błąd: Zamówienie o ID ' || p\_id\_zamowienia || ' nie istnieje.');

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Błąd: ' || SQLERRM);

END ZmienStatusZamowienia;

DECLARE

v\_id\_zamowienia INTEGER := 1;

v\_nowy\_status VARCHAR2(20) := 'Zrealizowane';

BEGIN

ZmienStatusZamowienia(v\_id\_zamowienia, v\_nowy\_status);

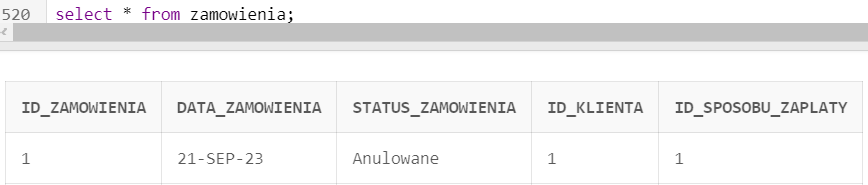
END;

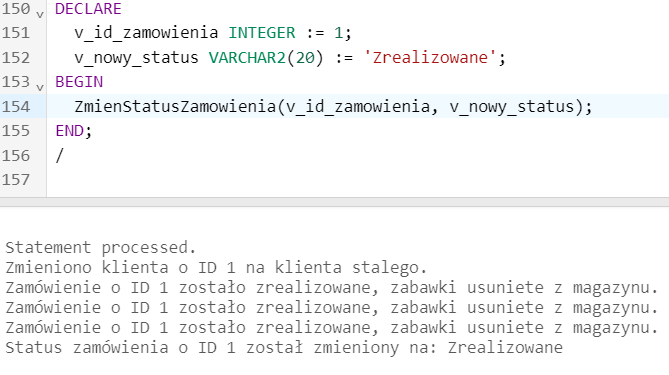
/

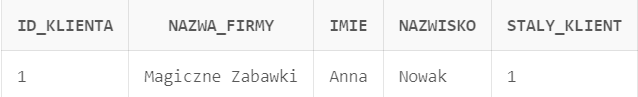
**PRZYKŁADY DZIAŁANIA PROCEDURY**

**POPRAWNE DZIAŁANIE**



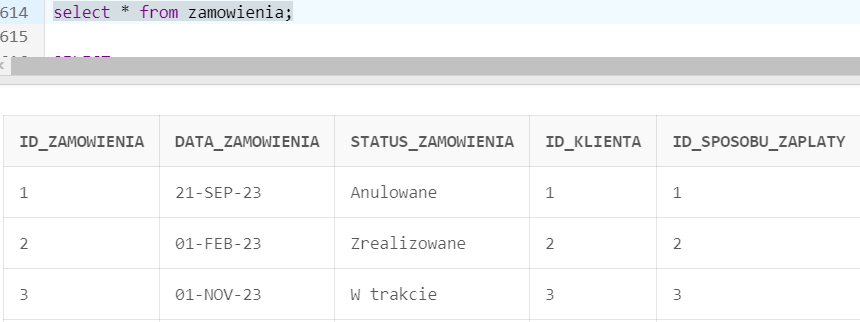


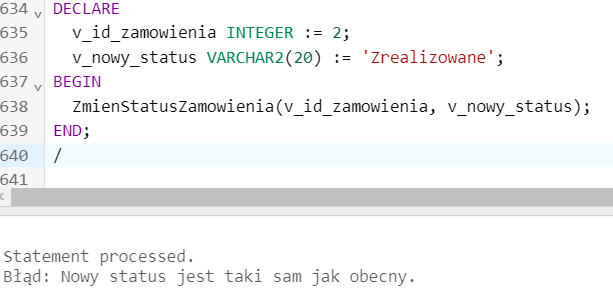




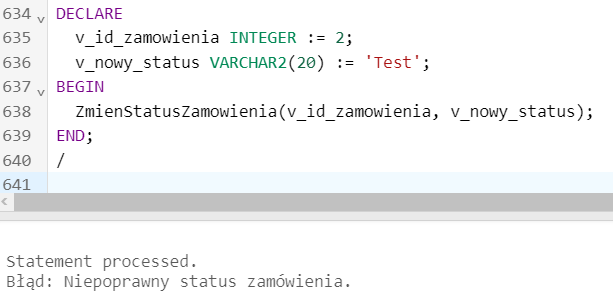
**BŁĘDY**

- Użycie tego samego statusu





- Użycie błędnego statusu



- Niewystarczająca liczba zabawek

