# Projekt

# Baza danych „handel towarami”

## Andrzej Białobrzeski

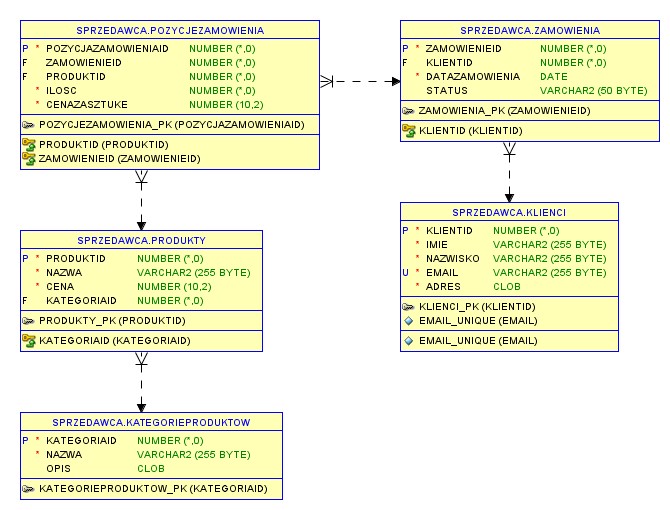
## Marek Dąbrowski

### **Założenia projektowe**

Podstawowymi założeniami były:

* Utworzenie struktury bazy danych reprezentującej wymianę handlową
* Stworzenie powiązań pomiędzy poszczególnymi elementami składowymi bazy danych
* Dodanie sekwencji umożliwiających numerowanie rekordów w tabelach
* Przygotowanie i ustawienie procedur, które automatycznie przy wprowadzaniu rekordów będą wywoływały dane sekwencje
* Dodanie ograniczeń dla wprowadzonych danych
* Wprowadzenie przykładów danych
* Stworzenie widoków w celu szybkiego przeglądania określonych danych
* Dodanie procedur realizujących operacje na tabelach
* Stworzenie funkcji ułatwiających wykonywanie zadań na bazie danych
* Przygotowanie dokumentacji projektu

### **Ogólny projekt systemu**



### **Kod SQL tworzący struktury bazodanowe**

1. **Tabele (min. 5 tabel): typy i wielkości pól, klucze główne i obce, ograniczenia,**

**wypełnić przykładowymi danymi (con. 15 rekordów)**

**Tabela KATEGORIEPRODUKTOW**

Utworzenie tabeli:

CREATE TABLE "SPRZEDAWCA"."KATEGORIEPRODUKTOW"

( "KATEGORIAID" NUMBER(\*,0) DEFAULT "SPRZEDAWCA"."KATEGORIE\_PRODUKTOW\_SEQ"."NEXTVAL",

"NAZWA" VARCHAR2(255 BYTE),

"OPIS" CLOB

)

Dodanie ograniczeń/klucza głównego:

ALTER TABLE "SPRZEDAWCA"."KATEGORIEPRODUKTOW" MODIFY ("NAZWA" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."KATEGORIEPRODUKTOW" ADD PRIMARY KEY ("KATEGORIAID")

Dodanie rekordów:

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('1','Sportowe');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('2','Spożywcze');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('3','Odzieżowe');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('4','Samochodowe');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('5','Wędkarskie');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('6','Meble');

Insert into SPRZEDAWCA.KATEGORIEPRODUKTOW (KATEGORIAID,NAZWA) values ('7','Sprzet elektryczny');

**Tabela Klienci**

Utworzenie tabeli:

CREATE TABLE "SPRZEDAWCA"."KLIENCI"

( "KLIENTID" NUMBER(\*,0) DEFAULT "SPRZEDAWCA"."KLIENCI\_SEQ"."NEXTVAL",

"IMIE" VARCHAR2(255 BYTE),

"NAZWISKO" VARCHAR2(255 BYTE),

"EMAIL" VARCHAR2(255 BYTE),

"ADRES" CLOB

)

Dodanie ograniczeń/klucza głównego:

ALTER TABLE "SPRZEDAWCA"."KLIENCI" MODIFY ("IMIE" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."KLIENCI" MODIFY ("NAZWISKO" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."KLIENCI" MODIFY ("EMAIL" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."KLIENCI" MODIFY ("ADRES" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."KLIENCI" ADD PRIMARY KEY ("KLIENTID")

ALTER TABLE "SPRZEDAWCA"."KLIENCI" ADD CONSTRAINT "EMAIL\_UNIQUE" UNIQUE ("EMAIL")

Dodanie rekordów:

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('1','Jan','Kowalski','Jan@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('2','Michał','Nowak','Michał@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('3','Kamil','Wielkopolski','Kamil@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('4','Mateusz','Koniecpolski','Mateusz@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('5','Andrzej','Rydz','Andrzej@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('6','Marek','Abramek','Marek@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('7','Wojciech','Abramowicz','Wojciech@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('8','Kamil','Abramiuk','Abramiuk@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('9','Bogusław','Abramowska','Bogusław@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('10','Ryszard','Achtelik','Ryszard@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('11','Krzysztof','Adach','Krzysztof@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('12','Patryk','Adamiuk','Patryk@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('13','Julia','Żywiołek','Julia@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('14','Agnieszka','Andrejczuk','Agnieszka@gmail.oom');

Insert into SPRZEDAWCA.KLIENCI (KLIENTID,IMIE,NAZWISKO,EMAIL) values ('15','Natalia','Andrusiuk','Natalia@gmail.oom');

**Tabela Pozycje Zamówienia**

Utworzenie tabeli:

CREATE TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA"

( "POZYCJAZAMOWIENIAID" NUMBER(\*,0) DEFAULT "SPRZEDAWCA"."POZYCJE\_ZAMOWIENIA\_SEQ"."NEXTVAL",

"ZAMOWIENIEID" NUMBER(\*,0),

"PRODUKTID" NUMBER(\*,0),

"ILOSC" NUMBER(\*,0),

"CENAZASZTUKE" NUMBER(10,2)

)

Dodanie ograniczeń/klucza głównego:

ALTER TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA" MODIFY ("ILOSC" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA" MODIFY ("CENAZASZTUKE" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA" ADD PRIMARY KEY ("POZYCJAZAMOWIENIAID")

Klucze obce:

ALTER TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA" ADD CONSTRAINT "ZAMOWIENIEID" FOREIGN KEY ("ZAMOWIENIEID")

REFERENCES "SPRZEDAWCA"."ZAMOWIENIA" ("ZAMOWIENIEID") ENABLE;

ALTER TABLE "SPRZEDAWCA"."POZYCJEZAMOWIENIA" ADD CONSTRAINT "PRODUKTID" FOREIGN KEY ("PRODUKTID")

REFERENCES "SPRZEDAWCA"."PRODUKTY" ("PRODUKTID") ENABLE;

Dodanie rekordów:

Insert into SPRZEDAWCA.POZYCJEZAMOWIENIA (POZYCJAZAMOWIENIAID,ZAMOWIENIEID,PRODUKTID,ILOSC,CENAZASZTUKE) values ('9','1','1','1','3000');

Insert into SPRZEDAWCA.POZYCJEZAMOWIENIA (POZYCJAZAMOWIENIAID,ZAMOWIENIEID,PRODUKTID,ILOSC,CENAZASZTUKE) values ('10','2','2','1','4000');

Insert into SPRZEDAWCA.POZYCJEZAMOWIENIA (POZYCJAZAMOWIENIAID,ZAMOWIENIEID,PRODUKTID,ILOSC,CENAZASZTUKE) values ('11','3','3','3','10');

Insert into SPRZEDAWCA.POZYCJEZAMOWIENIA (POZYCJAZAMOWIENIAID,ZAMOWIENIEID,PRODUKTID,ILOSC,CENAZASZTUKE) values ('12','4','4','2','15');

**Tabela Produkty**

Utworzenie tabeli:

CREATE TABLE "SPRZEDAWCA"."PRODUKTY"

( "PRODUKTID" NUMBER(\*,0) DEFAULT "SPRZEDAWCA"."PRODUKTY\_SEQ"."NEXTVAL",

"NAZWA" VARCHAR2(255 BYTE),

"CENA" NUMBER(10,2),

"KATEGORIAID" NUMBER(\*,0)

)

Dodanie ograniczeń/klucza głównego:

ALTER TABLE "SPRZEDAWCA"."PRODUKTY" MODIFY ("NAZWA" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."PRODUKTY" MODIFY ("CENA" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."PRODUKTY" ADD PRIMARY KEY ("PRODUKTID")

Klucze obce:

ALTER TABLE "SPRZEDAWCA"."PRODUKTY" ADD CONSTRAINT "KATEGORIAID" FOREIGN KEY ("KATEGORIAID")

REFERENCES "SPRZEDAWCA"."KATEGORIEPRODUKTOW" ("KATEGORIAID") ENABLE;

Dodanie rekordów:

Insert into SPRZEDAWCA.PRODUKTY (PRODUKTID,NAZWA,CENA,KATEGORIAID) values ('1','Łożko','3000','6');

Insert into SPRZEDAWCA.PRODUKTY (PRODUKTID,NAZWA,CENA,KATEGORIAID) values ('2','Laptop','4000','7');

Insert into SPRZEDAWCA.PRODUKTY (PRODUKTID,NAZWA,CENA,KATEGORIAID) values ('3','Frytki','10','2');

Insert into SPRZEDAWCA.PRODUKTY (PRODUKTID,NAZWA,CENA,KATEGORIAID) values ('4','Pizza','15','2');

Insert into SPRZEDAWCA.PRODUKTY (PRODUKTID,NAZWA,CENA,KATEGORIAID) values ('5','Buty Addidas','80','1');

**Tabela Zamówienia**

Utworzenie tabeli:

CREATE TABLE "SPRZEDAWCA"."ZAMOWIENIA"

( "ZAMOWIENIEID" NUMBER(\*,0) DEFAULT "SPRZEDAWCA"."ZAMOWIENIA\_SEQ"."NEXTVAL",

"KLIENTID" NUMBER(\*,0),

"DATAZAMOWIENIA" DATE,

"STATUS" VARCHAR2(50 BYTE)

)

Dodanie ograniczeń/klucza głównego:

ALTER TABLE "SPRZEDAWCA"."ZAMOWIENIA" MODIFY ("DATAZAMOWIENIA" NOT NULL ENABLE);

ALTER TABLE "SPRZEDAWCA"."ZAMOWIENIA" ADD PRIMARY KEY ("ZAMOWIENIEID")

Klucze obce:

ALTER TABLE "SPRZEDAWCA"."ZAMOWIENIA" ADD CONSTRAINT "KLIENTID" FOREIGN KEY ("KLIENTID")

REFERENCES "SPRZEDAWCA"."KLIENCI" ("KLIENTID") ENABLE;

Dodanie rekordów:

Insert into SPRZEDAWCA.ZAMOWIENIA (ZAMOWIENIEID,KLIENTID,DATAZAMOWIENIA,STATUS) values ('1','1',to\_date('24/05/06','RR/MM/DD'),'Zaakceptowane');

Insert into SPRZEDAWCA.ZAMOWIENIA (ZAMOWIENIEID,KLIENTID,DATAZAMOWIENIA,STATUS) values ('2','2',to\_date('24/05/09','RR/MM/DD'),'Odrzucone');

Insert into SPRZEDAWCA.ZAMOWIENIA (ZAMOWIENIEID,KLIENTID,DATAZAMOWIENIA,STATUS) values ('3','3',to\_date('24/05/14','RR/MM/DD'),'W trakcie');

Insert into SPRZEDAWCA.ZAMOWIENIA (ZAMOWIENIEID,KLIENTID,DATAZAMOWIENIA,STATUS) values ('4','4',to\_date('24/05/30','RR/MM/DD'),'Otwarte');

1. **Widoki**

Widok „WIDOKPRODUKTOWIKATEGORII”:

CREATE OR REPLACE FORCE EDITIONABLE VIEW "SPRZEDAWCA"."WIDOKPRODUKTOWIKATEGORII" ("PRODUKTID", "NAZWAPRODUKTU", "CENA", "NAZWAKATEGORII", "LICZBAZAMOWIEN") AS

SELECT

p.ProduktID,

p.Nazwa AS NazwaProduktu,

p.Cena,

kp.Nazwa AS NazwaKategorii,

COUNT(z.ZamowienieID) AS LiczbaZamowien

FROM

Produkty p

JOIN

KategorieProduktow kp ON p.KategoriaID = kp.KategoriaID

LEFT JOIN

PozycjeZamowienia pz ON p.ProduktID = pz.ProduktID

LEFT JOIN

Zamowienia z ON pz.ZamowienieID = z.ZamowienieID

GROUP BY

p.ProduktID, p.Nazwa, p.Cena, kp.Nazwa

;

Widok “WIDOKZAMOWIEN”

CREATE OR REPLACE FORCE EDITIONABLE VIEW "SPRZEDAWCA"."WIDOKZAMOWIEN" ("ZAMOWIENIEID", "DATAZAMOWIENIA", "STATUS", "PELNEIMIE", "NAZWAPRODUKTU", "ILOSC", "CENAZASZTUKE", "WARTOSCPOZYCJI") AS

SELECT

z.ZamowienieID,

z.DataZamowienia,

z.Status,

k.Imie || ' ' || k.Nazwisko AS PelneImie,

p.Nazwa AS NazwaProduktu,

pz.Ilosc,

pz.CenaZaSztuke,

(pz.Ilosc \* pz.CenaZaSztuke) AS WartoscPozycji

FROM

Zamowienia z

JOIN

Klienci k ON z.KlientID = k.KlientID

JOIN

PozycjeZamowienia pz ON z.ZamowienieID = pz.ZamowienieID

JOIN

Produkty p ON pz.ProduktID = p.ProduktID

;

### **Kod PL/SQL struktur programistycznych wraz z komentarzami:**

1. **Min. 4 funkcje: wykorzystać kursory, ROWTYPE, TYPE, wykorzystać instrukcje**

**warunkowe / pętle (na wyższą ocenę dodatkowo obsługa błędów);**

CREATE OR REPLACE EDITIONABLE PACKAGE "SPRZEDAWCA"."PAKIET\_KLIENCI" AS

TYPE ref\_cursor IS REF CURSOR;

FUNCTION pobierz\_klientow RETURN ref\_cursor;

END pakiet\_klienci;

/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "SPRZEDAWCA"."PAKIET\_KLIENCI" AS

FUNCTION pobierz\_klientow RETURN ref\_cursor IS

c\_klienci ref\_cursor;

BEGIN

OPEN c\_klienci FOR

SELECT \* FROM klienci;

RETURN c\_klienci;

EXCEPTION

WHEN OTHERS THEN

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

RAISE;

END pobierz\_klientow;

END pakiet\_klienci;

/

CREATE OR REPLACE EDITIONABLE PACKAGE "SPRZEDAWCA"."PAKIET\_PRODUKTY" AS

-- Typ referencyjny dla kursora

TYPE ref\_cursor IS REF CURSOR;

-- Funkcja zwracająca kursor z wybranymi danymi produktów

FUNCTION pobierz\_wybrane\_produkty RETURN ref\_cursor;

END pakiet\_produkty;

/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "SPRZEDAWCA"."PAKIET\_PRODUKTY" AS

FUNCTION pobierz\_wybrane\_produkty RETURN ref\_cursor IS

c\_produkty ref\_cursor;

BEGIN

OPEN c\_produkty FOR

SELECT pr.nazwa, pr.cena, kp.nazwa

FROM produkty pr

INNER JOIN KATEGORIEPRODUKTOW kp ON pr.kategoriaid = kp.kategoriaid;

RETURN c\_produkty;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nie znaleziono danych.');

RETURN c\_produkty; -- Zwróć pusty kursor

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Wystąpił nieoczekiwany błąd: ' || SQLERRM);

RAISE;

END pobierz\_wybrane\_produkty;

END pakiet\_produkty;

/

CREATE OR REPLACE EDITIONABLE PACKAGE "SPRZEDAWCA"."PAKIET\_ZAMOWIENIA" AS

TYPE ref\_cursor IS REF CURSOR;

FUNCTION pobierz\_zamowienia RETURN ref\_cursor;

FUNCTION pobierz\_pozycje\_zamowienia(zamowienie\_id IN INT) RETURN ref\_cursor;

END pakiet\_zamowienia;

/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "SPRZEDAWCA"."PAKIET\_PRODUKTY" AS

FUNCTION pobierz\_wybrane\_produkty RETURN ref\_cursor IS

c\_produkty ref\_cursor;

BEGIN

OPEN c\_produkty FOR

SELECT pr.nazwa, pr.cena, kp.nazwa

FROM produkty pr

INNER JOIN KATEGORIEPRODUKTOW kp ON pr.kategoriaid = kp.kategoriaid;

RETURN c\_produkty;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nie znaleziono danych.');

RETURN c\_produkty; -- Zwróć pusty kursor

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Wystąpił nieoczekiwany błąd: ' || SQLERRM);

RAISE;

END pobierz\_wybrane\_produkty;

END pakiet\_produkty;

/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "SPRZEDAWCA"."PAKIET\_ZAMOWIENIA" AS

FUNCTION pobierz\_zamowienia RETURN ref\_cursor IS

c\_zamowienia ref\_cursor;

BEGIN

OPEN c\_zamowienia FOR

SELECT \* FROM Zamowienia;

RETURN c\_zamowienia;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nie znaleziono danych.');

-- Zwróć pusty kursor, jeśli nie znaleziono danych

OPEN c\_zamowienia FOR SELECT \* FROM dual WHERE 1=0;

RETURN c\_zamowienia;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Wystąpił nieoczekiwany błąd: ' || SQLERRM);

-- Ponownie otwórz pusty kursor przed ponownym zgłoszeniem wyjątku

OPEN c\_zamowienia FOR SELECT \* FROM dual WHERE 1=0;

RETURN c\_zamowienia;

END pobierz\_zamowienia;

FUNCTION pobierz\_pozycje\_zamowienia(zamowienie\_id IN INT) RETURN ref\_cursor IS

c\_pozycje ref\_cursor;

BEGIN

OPEN c\_pozycje FOR

SELECT \* FROM PozycjeZamowienia WHERE ZamowienieID = zamowienie\_id;

RETURN c\_pozycje;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nie znaleziono pozycji dla zamówienia o ID: ' || zamowienie\_id);

-- Zwróć pusty kursor, jeśli nie znaleziono pozycji zamówienia

OPEN c\_pozycje FOR SELECT \* FROM dual WHERE 1=0;

RETURN c\_pozycje;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Wystąpił nieoczekiwany błąd: ' || SQLERRM);

-- Ponownie otwórz pusty kursor przed ponownym zgłoszeniem wyjątku

OPEN c\_pozycje FOR SELECT \* FROM dual WHERE 1=0;

RETURN c\_pozycje;

END pobierz\_pozycje\_zamowienia;

END pakiet\_zamowienia;

/

1. **Min. 4 procedury: wykorzystać kursory, ROWTYPE, TYPE, wykorzystać**

**instrukcje warunkowe / pętle (na wyższą ocenę dodatkowo obsługa błędów);**

ProceduraAKTUALIZUJCENEPRODUKTU:

CREATE OR REPLACE EDITIONABLE PROCEDURE "SPRZEDAWCA"."AKTUALIZUJCENEPRODUKTU" (p\_produktID IN Produkty.ProduktID%TYPE, p\_cena IN DECIMAL) IS

BEGIN

UPDATE Produkty SET Cena = p\_cena WHERE ProduktID = p\_produktID;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END AktualizujCeneProduktu;

/

ProceduraDODAJKATEGORIE:

CREATE OR REPLACE EDITIONABLE PROCEDURE "SPRZEDAWCA"."DODAJKATEGORIE" (p\_nazwa IN VARCHAR2, p\_opis IN VARCHAR2) IS

BEGIN

INSERT INTO KategorieProduktow (Nazwa, Opis)

VALUES (p\_nazwa, p\_opis);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END DodajKategorie;

/

ProceduraUTWORZZAMOWIENIE:

CREATE OR REPLACE EDITIONABLE PROCEDURE "SPRZEDAWCA"."UTWORZZAMOWIENIE" (p\_klientID IN Klienci.KlientID%TYPE, p\_dataZamowienia IN DATE) IS

v\_zamowienieID Zamowienia.ZamowienieID%TYPE;

BEGIN

INSERT INTO Zamowienia (KlientID, DataZamowienia, Status)

VALUES (p\_klientID, p\_dataZamowienia, 'Nowe')

RETURNING ZamowienieID INTO v\_zamowienieID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Utworzono zamówienie o ID: ' || v\_zamowienieID);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END UtworzZamowienie;

ProceduraWYSWIETLPRODUKTYKATEGORII:

CREATE OR REPLACE EDITIONABLE PROCEDURE "SPRZEDAWCA"."WYSWIETLPRODUKTYKATEGORII" (p\_kategoriaID IN KategorieProduktow.KategoriaID%TYPE) IS

CURSOR c\_produkty IS

SELECT Nazwa, Cena FROM Produkty WHERE KategoriaID = p\_kategoriaID;

v\_nazwa Produkty.Nazwa%TYPE;

v\_cena Produkty.Cena%TYPE;

BEGIN

OPEN c\_produkty;

LOOP

FETCH c\_produkty INTO v\_nazwa, v\_cena;

EXIT WHEN c\_produkty%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Produkt: ' || v\_nazwa || ', Cena: ' || v\_cena);

END LOOP;

CLOSE c\_produkty;

EXCEPTION

WHEN OTHERS THEN

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

END WyswietlProduktyKategorii;

/

1. **Wykorzystać sekwencje do inkrementacji;**

Tabela kategorie produktów:

CREATE SEQUENCE "SPRZEDAWCA"."KATEGORIE\_PRODUKTOW\_SEQ" MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER CYCLE NOKEEP NOSCALE GLOBAL ;

CREATE OR REPLACE EDITIONABLE TRIGGER "SPRZEDAWCA"."TRG\_KATEGORIE\_PRODUKTOW"

BEFORE INSERT OR UPDATE ON kategorieproduktow

FOR EACH ROW

BEGIN

:NEW.KATEGORIAID := KATEGORIE\_PRODUKTOW\_seq.NEXTVAL;

END;

/

ALTER TRIGGER "SPRZEDAWCA"."TRG\_KATEGORIE\_PRODUKTOW" ENABLE;

Tabela klienci:

CREATE SEQUENCE "SPRZEDAWCA"."KLIENCI\_SEQ" MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER CYCLE NOKEEP NOSCALE GLOBAL ;

CREATE OR REPLACE EDITIONABLE TRIGGER "SPRZEDAWCA"."TRG\_KLIENCI"

BEFORE INSERT OR UPDATE ON klienci

FOR EACH ROW

BEGIN

:NEW.KLIENTID := KLIENCI\_seq.NEXTVAL;

END;

/

ALTER TRIGGER "SPRZEDAWCA"."TRG\_KLIENCI" ENABLE;

Tabela pozycje zamówienia:

CREATE SEQUENCE "SPRZEDAWCA"."POZYCJE\_ZAMOWIENIA\_SEQ" MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER CYCLE NOKEEP NOSCALE GLOBAL ;

CREATE OR REPLACE EDITIONABLE TRIGGER "SPRZEDAWCA"."TRG\_POZYCJE\_ZAMOWIENIA"

BEFORE INSERT OR UPDATE ON pozycjezamowienia

FOR EACH ROW

BEGIN

:NEW.pozycjazamowieniaid := POZYCJE\_ZAMOWIENIA\_seq.NEXTVAL;

END;

/

ALTER TRIGGER "SPRZEDAWCA"."TRG\_POZYCJE\_ZAMOWIENIA" ENABLE;

Tabela produkty:

CREATE SEQUENCE "SPRZEDAWCA"."PRODUKTY\_SEQ" MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER CYCLE NOKEEP NOSCALE GLOBAL ;

CREATE OR REPLACE EDITIONABLE TRIGGER "SPRZEDAWCA"."TRG\_PRODUKTY"

BEFORE INSERT OR UPDATE ON produkty

FOR EACH ROW

BEGIN

:NEW.produktid := produkty\_seq.NEXTVAL;

END;

/

ALTER TRIGGER "SPRZEDAWCA"."TRG\_PRODUKTY" ENABLE;

Tabela zamówienia:

CREATE SEQUENCE "SPRZEDAWCA"."ZAMOWIENIA\_SEQ" MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER CYCLE NOKEEP NOSCALE GLOBAL ;

CREATE OR REPLACE EDITIONABLE TRIGGER "SPRZEDAWCA"."TRG\_ZAMOWIENIA"

BEFORE INSERT OR UPDATE ON zamowienia

FOR EACH ROW

BEGIN

:NEW.zamowienieid := zamowienia\_seq.NEXTVAL;

END;

/

ALTER TRIGGER "SPRZEDAWCA"."TRG\_ZAMOWIENIA" ENABLE;

**d)** **Min. 4 wyzwalacze (nie do inkrementacji).**