Magdalena Wiśniewska

**Baza danych sklepu internetowego sklepu z aranżacją wnętrz**

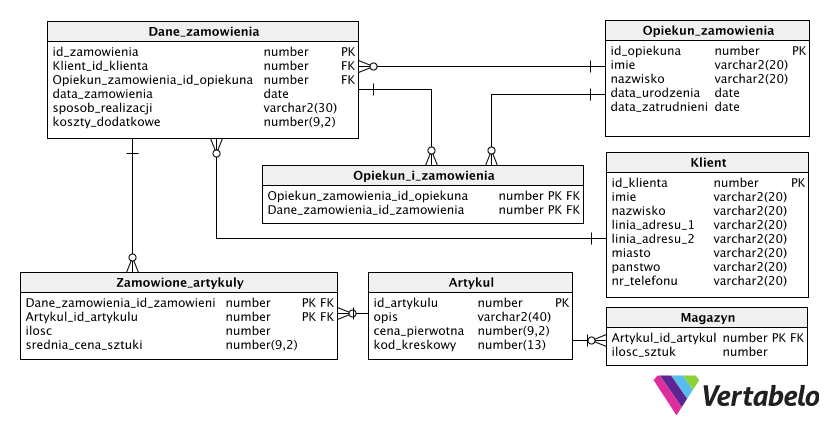
1. **OPIS**

Moja baza danych będzie przechowywać informacje dotyczące zamówień realizowanych przez luksusowy sklep internetowy. Ze względu na to, ze kwoty zamówienia bywają zazwyczaj dosyć wysokie, a przesyłki są zazwyczaj wielkogabarytowe, każde z zamówień ma swojego opiekuna, który czuwa nad tym, żeby proces dostarczenia go pod wskazany adres przebiegał bez zakłóceń. Każde zamówienie ma jednego opiekuna, ale ten sam opiekun czuwa nad kilkoma zamówieniami. Istotne jest to, ze czasami w trakcie realizacji zamówienia istnieje konieczność zmiany jego opiekuna, ale zapamiętujemy nazwiska wszystkich, którzy pracowali nad realizaja zamówienia. Zakładam, ze każdy z klientów (zarówno z Polski i z zagranicy) może realizować dowolnie wiele zamówień. W ramach każdego zamówienia może się znaleźć dowolnie wiele artykułów, ale ten sam artykuł może występować w różnych zamówieniach (w ramach jednego zamówienia ta sama pozycja może znaleźć się wiele razy). Baza będzie zawierała tez informacje o aktualnych stanach magazynowych.

1. **TABELE W BAZIE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Klient**:   * **id\_klienta** * imi * nazwisko * liniaadresu\_1 * liniaadresu\_2 * miasto * panstwo * nr\_telefonu | **Opiekun\_zamowienia:**   * **id\_opiekuna** * imie * nazwisko * data\_urodzenia * data\_zatrudnienia | | **Dane\_ zamowienia:**   * **id\_zamowienia** * **id\_klienta** * **id\_opiekuna** * data\_zamowienia * sposob\_realizacji * koszty\_dodatkowe | |
| **Artykul:**   * **id\_artykułu** * opis * cena\_pierwotna * kod\_kreskowy | **Zamowione\_artykuly:**   * **id\_zamowienia** * **id\_artykulu** * ilosc * srednia\_cena\_sztuki | **Magazyn:**   * **id\_artykułu** * ilosc\_sztuk | | **Opiekun\_i\_zamowienia:**   * **id\_zamowienia**   **id\_opiekuna.** |

1. **WYKRES ERD**



1. **DDL (Data Definition Language):**

**/\* ------------------------------------Klient---------------- \***

CREATE TABLE Klient

(id\_klienta NUMBER primary key,

imie VARCHAR2(20) NOT NULL ,

nazwisko VARCHAR2(20) NOT NULL,

liniaadresu\_1 VARCHAR2(20) NOT NULL,

liniaadresu\_2 VARCHAR2(20) NOT NULL,

miasto VARCHAR2(20) NOT NULL,

panstwo VARCHAR2(20) NOT NULL,

nr\_telefonu VARCHAR2(20) NOT NULL);

**/\* ------------------------------Opiekun\_zamowienia---------- \*/**

CREATE TABLE Opiekun\_zamowienia

(id\_opiekuna NUMBER primary key,

imie VARCHAR2(20) NOT NULL,

nazwisko VARCHAR2(20) NOT NULL,

data\_urodzenia DATE NOT NULL,

data\_zatrudnienia DATE NOT NULL);

**/\* -------------------------------Dane\_zamowienia-----------** \*/

CREATE TABLE Dane\_zamowienia

(id\_zamowienia NUMBER primary key,

id\_klienta NUMBER NOT NULL references Klient(id\_klienta),

id\_opiekuna NUMBER NOT NULL references Opiekun\_zamowienia(id\_opiekuna),

data\_zamówienia DATE NOT NULL,

sposob\_realizacji VARCHAR2(30) NOT NULL,

koszty\_dodatkowe NUMBER(9,2) NOT NULL);

**/\* ------------------------------------Artykul--------------- \*/**

CREATE TABLE Artykul

(id\_artykulu NUMBER primary key,

opis VARCHAR2(40) NOT NULL,

cena\_pierwotna NUMBER(9,2) NOT NULL,

kod\_kreskowy VARCHAR2(13) NOT NULL);

**/\* ------------------------------Zamowione\_artykuly---------- \*/**

CREATE TABLE Zamowione\_artykuly

(id\_zamowienia NUMBER NOT NULL references Dane\_zamowienia(id\_zamowienia),

id\_artykulu NUMBER NOT NULL references Artykul(id\_artykulu),

ilosc NUMBER NOT NULL,

srednia\_cena\_sztuki NUMBER(9,2) NOT NULL);

**/\* ------------------------------------Magazyn--------------- \*/**

CREATE TABLE Magazyn

(id\_artykulu NUMBER NOT NULL references Artykul(id\_artykulu),

ilosc\_sztuk NUMBER NOT NULL);

**/\* -----------------------------Opiekun\_i\_zamowienia--------- \*/**

CREATE TABLE Opiekun\_i\_zamowienia

(id\_zamowienia NUMBER NOT NULL references Dane\_zamowienia(id\_zamowienia),

id\_opiekuna NUMBER NOT NULL references Opiekun\_zamowienia(id\_opiekuna));

1. **DML(Data Manipulation Language)**:

**/\* ------------------------------------Klient---------------- \*/**

INSERT INTO Klient VALUES (1,'Elzbieta','Kot','Kowaliowa 67','08-400','Garwolin','Polska','+48509765345');

INSERT INTO Klient VALUES (2,'Aleksandra','Urlich','Lolka 59/74','07-360','Warszawa','Polska','+48601764025');

INSERT INTO Klient VALUES (3,'Witold','Zbyrasz','Trybuny 4','04-945','Poznan','Polska','+48500627645');

INSERT INTO Klient VALUES (4,'John','Zet','Seifertova 3','130 00','Praga','Czechy','+4201764025');

**/\* ------------------------------Opiekun\_zamowienia---------- \*/**

INSERT INTO Opiekun\_zamowienia VALUES(1,'Jan','Kulczyk','66/05/14','15/06/09');

INSERT INTO Opiekun\_zamowienia VALUES (2,'Maria','Wider','78/03/10','15/06/09');

INSERT INTO Opiekun\_zamowienia VALUES (3,'Arkadiusz','Lach','84/12/27','15/08/11');

INSERT INTO Opiekun\_zamowienia VALUES (4,'Ewa','Salen','86/10/04','15/08/23');

**/\* -------------------------------Dane\_zamowienia----------- \*/**

INSERT INTO Dane\_zamowienia VALUES (1,1,1,'15/06/15','odbior wlasny',5.0);

INSERT INTO Dane\_zamowienia VALUES (2,2,2,'15/06/22','przesylka kurierska DHL',20.5);

INSERT INTO Dane\_zamowienia VALUES (3,1,2,'15/07/01','odbior wlasny',15.0);

INSERT INTO Dane\_zamowienia VALUES (4,1,1,'15/07/11','odbior wlasny',10.0);

INSERT INTO Dane\_zamowienia VALUES (5,3,2,'15/08/02','przesylka kurierska DHL',20.50);

INSERT INTO Dane\_zamowienia VALUES (6,4,3,'15/08/13','przesylka kurierska TNT',240.50);

INSERT INTO Dane\_zamowienia VALUES (7,3,3,'15/08/18','przesylka kurierska DHL',20.50);

INSERT INTO Dane\_zamowienia VALUES (8,4,4,'15/08/25','przesylka kurierska TNT',70.5);

**/\* ------------------------------------Artykul--------------- \*/**

INSERT INTO Artykul VALUES (1,'wazon szklany pieciokat jasny roz',199.50,5901234123445);

INSERT INTO Artykul VALUES (2,'wazon drewniany kwadrat blekit',219.50,5901234123435);

INSERT INTO Artykul VALUES (3,'figurka aniol biel',89.50,5901234143986);

INSERT INTO Artykul VALUES (4,'zyrandol medium zloto',4699.50,5059012300058);

INSERT INTO Artykul VALUES (5,'fotel muszla turkus',3299.50,5901234155553);

**/\* ------------------------------Zamowione\_artykuly---------- \*/**

INSERT INTO Zamowione\_artykuly VALUES (1,1,3,190.0);

INSERT INTO Zamowione\_artykuly VALUES (1,2,2,215.0);

INSERT INTO Zamowione\_artykuly VALUES (2,2,2,219.5);

INSERT INTO Zamowione\_artykuly VALUES (3,5,2,3200.0);

INSERT INTO Zamowione\_artykuly VALUES (4,4,1,699.5);

INSERT INTO Zamowione\_artykuly VALUES (5,3,1,89.5);

INSERT INTO Zamowione\_artykuly VALUES (6,4,1,6999.50);

INSERT INTO Zamowione\_artykuly VALUES (7,1,2,195.0);

INSERT INTO Zamowione\_artykuly VALUES (8,3,2,85.0);

**/\* ------------------------------------Magazyn--------------- \*/**

INSERT INTO Magazyn VALUES (1,32);

INSERT INTO Magazyn VALUES (2,54);

INSERT INTO Magazyn VALUES (3,43);

INSERT INTO Magazyn VALUES (4,4);

INSERT INTO Magazyn VALUES (5,6);

**/\* -----------------------------Opiekun\_i\_zamowienia--------- \*/**

INSERT INTO Opiekun\_i\_zamowienia VALUES (1,1);

INSERT INTO Opiekun\_i\_zamowienia VALUES (2,1);

INSERT INTO Opiekun\_i\_zamowienia VALUES (2,2);

INSERT INTO Opiekun\_i\_zamowienia VALUES (3,2);

INSERT INTO Opiekun\_i\_zamowienia VALUES (4,1);

INSERT INTO Opiekun\_i\_zamowienia VALUES (5,2);

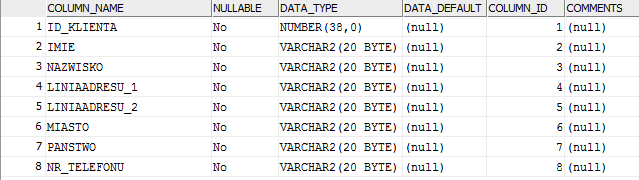
INSERT INTO Opiekun\_i\_zamowienia VALUES (6,3);

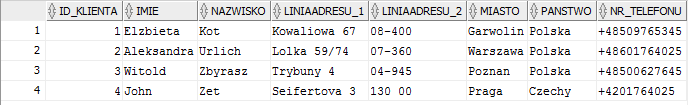
INSERT INTO Opiekun\_i\_zamowienia VALUES (7,3);

INSERT INTO Opiekun\_i\_zamowienia VALUES (8,4);

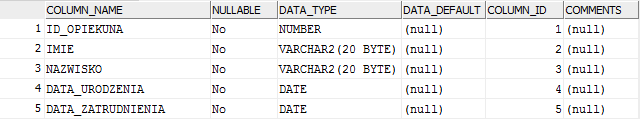
INSERT INTO Opiekun\_i\_zamowienia VALUES (8,3);

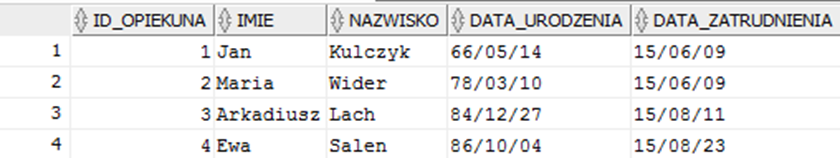
1. **TABELE Z SQL DEVELOPERA**
2. **Klient**



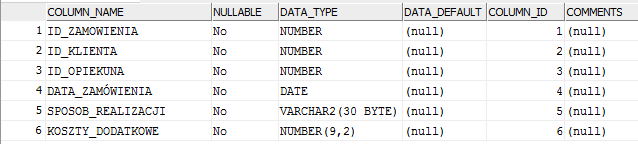


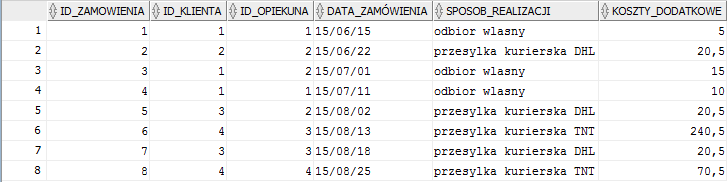
1. **Opiekun\_zamowienia**



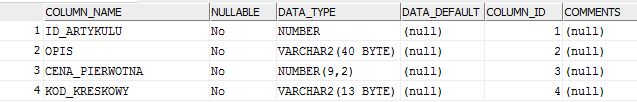
****

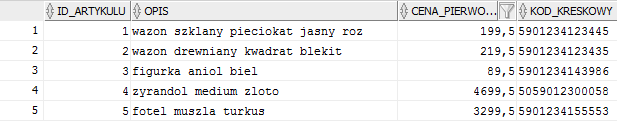
1. **Dane\_zamowienia**



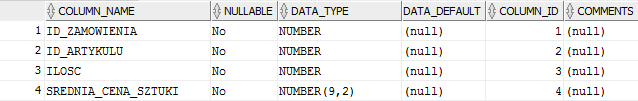


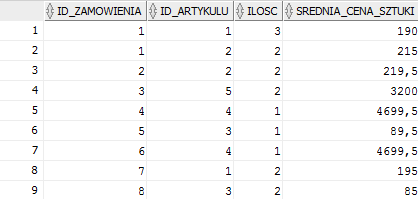
1. **Artykul**



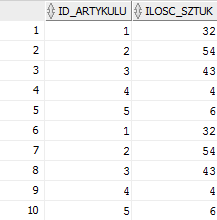


1. **Zamowione\_artykuly**





1. **Magazyn**

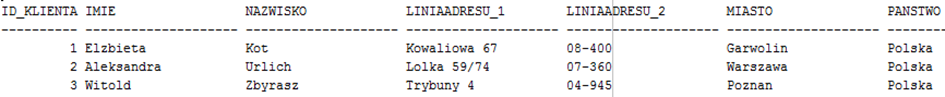


1. **Opiekun\_i\_zamowienia**



1. **ZAPYTANIA SQL:**
2. **z jednej tabeli z warunkami Where:**
3. Wyświetl dane zamówień realizowanych w Polsce

SELECT \* FROM Klient WHERE panstwo=’Polska’;



1. **z wielu tabel z warunkami Where:**
2. Wyświetl imiona i nazwiska osób, które są aktualnymi opiekunami konkretnych zamówień.

SELECT

d.id\_zamowienia

, o.imie, o.nazwisko

FROM

Dane\_zamowienia d

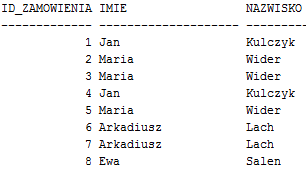
, Opiekun\_zamowienia o

WHERE

d.id\_opiekuna=o.id\_opiekuna

ORDER BY

d.id\_zamowienia;



1. Wyświetl id\_zamowienia wraz z suma do zaplaty za wszystkie artykuly (bez kosztów dodatkowych posortowane rosnąco zdodnie z id\_zamowienia

SELECT

d.id\_zamowienia

, sum (z.SREDNIA\_CENA\_SZTUKI\*z.ILOSC)Cena\_artykulow

FROM

Dane\_zamowienia d

, Zamowione\_artykuly z

WHERE

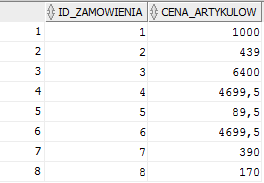
d.id\_zamowienia=z.id\_zamowienia

GROUP BY

d.ID\_ZAMOWIENIA

ORDER BY

d.ID\_ZAMOWIENIA;



1. Wyświetl nazwe opis artykulu i obecny jego stan magazynowy .

SELECT

a.opis

,m.ilosc\_sztuk

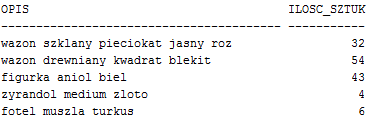
FROM

Artykul a

,Magazyn m

WHERE

a.id\_artykulu=m.id\_artykulu;



1. **grupujące na jednej tabeli:**
2. Oblicz ile sztuk każdego z artykułów wystąpiło w sumie we wszystkich zamówieniach

SELECT

id\_artykulu

, SUM(ilosc)

FROM

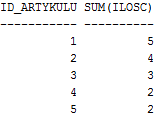
Zamowione\_artykuly

GROUP BY

id\_artykulu

ORDER BY

id\_artykulu;



1. Oblicz średnia kosztów dodatkowych zamówień od sierpnia 2015

SELECT

AVG (koszty\_dodatkowe) Srednia

FROM

Dane\_zamowienia

WHERE

Data\_zamówienia>'15/07/31';



1. **grupujące na jednej tabeli z warunkiem Having**
2. Wybierz te id\_klienta, gdzie były minimum 3 zamówienia

SELECT

id\_klienta

, count(\*)

FROM

Dane\_zamowienia

GROUP By

id\_klienta

HAVING

count(\*)>2;



1. Podaj zamówienia, którym w trakcie realizacji zmienił się opiekun.

SELECT

id\_zamowienia

FROM

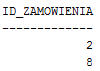
Opiekun\_i\_zamowienia

GROUP BY

id\_zamowienia

HAVING

COUNT (\*)>1;



1. **grupujące na wielu tabelach z warunkami where i having**
2. Wybierz te id\_klienta, gdzie były minimum 2 zamówienia i ich opiekunem aktualnym nie jest Arkadiusz Lach

SELECT

d.id\_klienta

FROM

Dane\_zamowienia d

,Opiekun\_zamowienia o

WHERE

d.id\_opiekuna=o.id\_opiekuna

AND o.imie NOT LIKE 'Arkadiusz'

AND o.nazwisko NOT LIKE 'Lach'

GROUP BY

id\_klienta

HAVING

count(\*)>1;



1. Podaj zamówienia z zagranicy, którym w trakcie realizacji zmienił się opiekun.

SELECT

op.id\_zamowienia

FROM

Opiekun\_i\_zamowienia op

, Klient k

, Dane\_zamowienia d

WHERE

d.id\_klienta=k.id\_klienta

AND d.id\_opiekuna=op.id\_opiekuna

AND k.panstwo NOT LIKE 'Polska'

GROUP BY

op.id\_zamowienia

HAVING COUNT (\*)>1;



1. Podaj id klientów i zamówień wraz z sumą zamówionych artykułów, gdzie przy okazji jednego zamówienia znazło się więcej niż jedna sztuka artykułu

SELECT

d.id\_klienta

,z.id\_zamowienia

, SUM(z.ilosc)

FROM

Dane\_zamowienia d

, Zamowione\_artykuly z

WHERE

z.id\_zamowienia=d.id\_zamowienia

GROUP BY

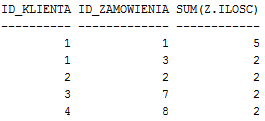
d.id\_klienta,z.id\_zamowienia

HAVING

SUM(z.ilosc)>1

ORDER BY

d.id\_klienta;



1. **podzapytania zwykłe w warunku where**
2. Wyświetl id\_zamowienia wraz z suma do zaplaty za wszystkie artykuly (bez kosztów dodatkowych w tych zamówieniach, którym nie zmienial się opiekun

SELECT

d.id\_zamowienia

, sum(z.SREDNIA\_CENA\_SZTUKI\*z.ILOSC)Cena\_artykulow

FROM

Dane\_zamowienia d, Zamowione\_artykuly z

WHERE

d.id\_zamowienia=z.id\_zamowienia

AND d.ID\_ZAMOWIENIA NOT IN

(SELECT id\_zamowienia

FROM Opiekun\_i\_zamowienia

GROUP BY id\_zamowienia

HAVING COUNT (\*)>1)

GROUP BY

d.ID\_ZAMOWIENIA

ORDER BY

d.ID\_ZAMOWIENIA;



1. Wyświetl id\_zamowienia wraz z ostatnim opiekunem zamowien, w których sposobem realizacji był „odbior wlasny”

SELECT

z.ID\_ZAMOWIENIA

, o.IMIE

, o.NAZWISKO

FROM

Opiekun\_zamowienia o

, Opiekun\_i\_zamowienia z

WHERE

o.ID\_OPIEKUNA=z.ID\_OPIEKUNA

AND z.ID\_ZAMOWIENIA IN

(SELECT ID\_ZAMOWIENIA

FROM DANE\_ZAMOWIENIA

WHERE SPOSOB\_REALIZACJI LIKE 'odbior wlasny');



1. **podzapytanie zwykłe z podzapytaniem w klauzuli from**
2. Wyświetl id\_zamowienia wraz z suma do zaplaty za wszystkie artykuly uwzgledniajac koszty dodatkowe sortując malejąco wg calosci do zaplaty

SELECT

d.id\_zamowienia

, (n.cena\_artykulow+d.koszty\_dodatkowe)Calosc\_do\_zaplaty

FROM

Dane\_zamowienia d,

(SELECT a.id\_zamowienia, sum(b.SREDNIA\_CENA\_SZTUKI\*b.ILOSC)cena\_artykulow

FROM Dane\_zamowienia a, Zamowione\_artykuly b

WHERE a.id\_zamowienia=b.id\_zamowienia

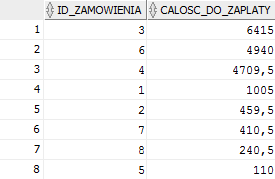
GROUP a.ID\_ZAMOWIENIA) n

WHERE

d.ID\_ZAMOWIENIA=n.id\_zamowienia

ORDER BY

(n.cena\_artykulow+d.koszty\_dodatkowe) DESC;



1. Wyświetl sumy do zaplaty wszystkich zamowien zamowionych przez poszczegolnych klientow

SELECT

k.NAZWISKO

,k.IMIE

,SUM(n.CALOSC\_DO\_ZAPLATY)

FROM

Klient k

,DANE\_ZAMOWIENIA d,

(SELECT d.id\_zamowienia,(n.cena\_artykulow+d.koszty\_dodatkowe)Calosc\_do\_zaplaty

FROM Dane\_zamowienia d,

(SELECT a.id\_zamowienia, sum(b.SREDNIA\_CENA\_SZTUKI\*b.ILOSC)cena\_artykulow

FROM Dane\_zamowienia a, Zamowione\_artykuly b

WHERE a.id\_zamowienia=b.id\_zamowienia

GROUP BY a.ID\_ZAMOWIENIA) n

WHERE d.ID\_ZAMOWIENIA=n.id\_zamowienia)s

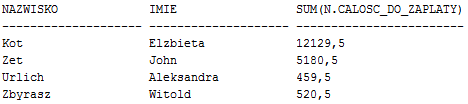
WHERE

k.ID\_KLIENTA=d.ID\_KLIENTA

AND d.ID\_ZAMOWIENIA=n.id\_zamowienia

GROUP BY

k.NAZWISKO, k.IMIE;



1. **podzapytania zwykłe w warunku having**
2. Wyświetl te id\_zamowien wraz ze imieniem i nazwiskiem skladajacych je klientów, gdzie wśród zamówionych artykulow znalazło się najwięcej wazów

SELECT

pom.id

, k.NAZWISKO

,k.IMIE

, pom.wazony

FROM

Klient k,

(SELECT za.ID\_ZAMOWIENIA id, sum(za.ilosc)wazony

FROM Zamowione\_artykuly za, ARTYKUL a

WHERE za.ID\_ARTYKULU=a.ID\_ARTYKULU AND a.OPIS LIKE '%wazon%'

HAVING SUM(za.ilosc)=

(SELECTmax(s.suma)from

(SELECT za.ID\_ZAMOWIENIA, sum(za.ilosc)suma

FROM Zamowione\_artykuly za , ARTYKUL a

WHERE za.ID\_ARTYKULU=a.ID\_ARTYKULU

AND a.OPIS LIKE '%wazon%'

GROUP BY za.ID\_ZAMOWIENIA)s )

GROUP BY za.ID\_ZAMOWIENIA )pom

WHERE

pom.id=k.ID\_KLIENTA;



1. Wyświetl id, opis i sume artykułów danego typu, które znalazły się łącznie we wszystkich zamówieniach, których suma wynosiła tyle samo, ile suma łącznych zamówień produktu o kodzie kreskowym '5059012300058'. Nie wyświetlaj danych na temat artykułu o tym kodzie kreskowym.

SELECT

z.id\_artykulu

,a.opis, SUM(z.ilosc)suma

FROM

Zamowione\_artykuly z

, Artykul a

WHERE

z.ID\_ARTYKULU = a.ID\_ARTYKULU

AND a.KOD\_KRESKOWY not like '5059012300058'

HAVING

SUM(z.ilosc)=(select suma from

(SELECT z.id\_artykulu, SUM(z.ilosc)suma

FROM Zamowione\_artykuly z, Artykul a

where a.kod\_kreskowy LIKE '5059012300058'

and z.ID\_ARTYKULU = a.ID\_ARTYKULU

GROUP BY z.id\_artykulu))

GROUP BY z.id\_artykulu, a.opis;



1. **podzapytania skorelowane z podzapytaniem w klauzuli where**
2. Wyświetl 2 najdluzej zatrudnione osoby w firmie wraz z iloscia zamowien, z którymi kiedykolwiek mieli do czynienia

SELECT

o.IMIE

, o.NAZWISKO

, o.DATA\_ZATRUDNIENIA

, p.ILOSC\_ZAMOWIEN

FROM

Opiekun\_zamowienia o,

(SELECT id\_opiekuna, COUNT(\*)ILOSC\_ZAMOWIEN

FROM OPIEKUN\_I\_ZAMOWIENIA

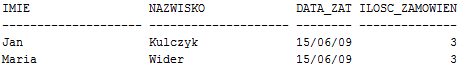
GROUP BY ID\_OPIEKUNA) p

WHERE o.id\_opiekuna=p.id\_opiekuna AND 2>

(SELECT count(\*)

FROM OPIEKUN\_ZAMOWIENIA z

WHERE o.DATA\_ZATRUDNIENIA > z.data\_zatrudnienia);



1. Wyświetl id zamówień, w których znalazły się figurki wraz z wartością ich średniej ceny za sztukę, w których średnia cena za jej nabycie była najnizsza.

SELECT

za.id\_zamowienia id

, za.SREDNIA\_CENA\_SZTUKI cena

FROM

zamowione\_artykuly za

WHERE

za.SREDNIA\_CENA\_SZTUKI=

(SELECT MIN (cena)from

(SELECT za.id\_zamowienia id, za.SREDNIA\_CENA\_SZTUKI cena

FROM zamowione\_artykuly za

WHERE EXISTS

(SELECT 'x' FROM artykul a

WHERE opis LIKE '%figurka%'

AND za.id\_artykulu=a.id\_artykulu)));



1. **Podzapytanie skorelowane z podzapytaniem w warunku having**
2. Wyświetl id zamówienia, miesiąc w którym zostało dokonane oraz cene do zapłaty za artykuły (bez kosztów dodatkowych), w przypadku których ta cena jest nizsza od sredniej ceny do zapłaty za zamówienia dokonane w tym samym miesiącu, w którym to zamówienie zostało złozone

SELECT

pom.miesiac

, pom.id\_zamowienia

, SUM (z.SREDNIA\_CENA\_SZTUKI\*z.ILOSC)

FROM

zamowione\_artykuly z,

(SELECT id\_zamowienia

,EXTRACT(month FROM data\_zamówienia)miesiac

FROM DANE\_ZAMOWIENIA)pom

WHERE pom.ID\_ZAMOWIENIA=z.ID\_ZAMOWIENIA

HAVING SUM (z.SREDNIA\_CENA\_SZTUKI\*z.ILOSC)<

(SELECT m.srednia from

(SELECT mop.miesiac mies

,avg(zam.SREDNIA\_CENA\_SZTUKI\*zam.ILOSC)srednia

FROM

(SELECT id\_zamowienia id\_zamowienia,

EXTRACT(month FROM data\_zamówienia)miesiac

FROM DANE\_ZAMOWIENIA)mop,zamowione\_artykuly zam

WHERE mop.ID\_ZAMOWIENIA=zam.ID\_ZAMOWIENIA

GROUP BY mop.miesiac)m

WHERE m.mies=pom.miesiac)

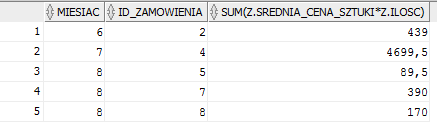
GROUP BY

pom.miesiac

, pom.id\_zamowienia

ORDER BY

pom.id\_zamowienia;



1. **PL SQL**
2. **Procedura  z Kursorem**
3. Podwyzsz o 10% ceny artykułom, które kosztują poniżej 200 zł i obniż cene tych, których cena wynosi więcej niż 4000 zł

**/\* -------------------------------Dane\_zamowienia-----------** \*/

CREATE TABLE HIST\_CEN

(LP NUMBER primary key,

ID\_ARTYKULU references ARTYKUL (id\_artykulu),

STARA NUMBER NOT NULL,

NOWA NUMBER NOT NULL,

DATA DATE NOT NULL );

CREATE OR REPLACE PROCEDURE

zmien\_ceny (MaxCena NUMBER, MinCena NUMBER)

AS

stara\_cena artykul.cena\_pierwotna%type;

nowa\_cena artykul.cena\_pierwotna%TYPE;

NR artykul.id\_artykulu%TYPE;

Info VARCHAR2(300);

lp hist\_cen.lp%type;

CURSOR

artykuly IS SELECT id\_artykulu, opis, NVL(cena\_pierwotna, 0)cena

FROM

artykul;

art artykuly%Rowtype;

BEGIN

OPEN artykuly;

LOOP

FETCH artykuly INTO art;

EXIT WHEN artykuly%NOTFOUND;

stara\_cena := art.cena;

nowa\_cena := art.cena;

IF nowa\_cena <= MinCena THEN

nowa\_cena := nowa\_cena \* 1.1;

Info := 'Artykul o id ' || art.id\_artykulu || ' i opisie '||art.opis|| ' ma podwyzszona cene z '||stara\_cena||' zl do ' || nowa\_cena ||' zl';

END IF;

IF nowa\_cena > MaxCena THEN

nowa\_cena := nowa\_cena \* 0.9;

Info := 'Artykul o id ' || art.id\_artykulu || ' i opisie '||art.opis|| ' ma obnizona cene z '||stara\_cena||' zl do ' || nowa\_cena ||' zl';

END IF;

IF stara\_cena <> nowa\_cena THEN

UPDATE artykul SET CENA\_PIERWOTNA = nowa\_cena WHERE id\_artykulu=art.id\_artykulu;

SELECT MAX(lp) into lp from hist\_cen;

insert into hist\_cen values (nvl(lp,0)+1,art.id\_artykulu, stara\_cena,nowa\_cena,sysdate);

dbms\_output.put\_line (Info);

END IF;

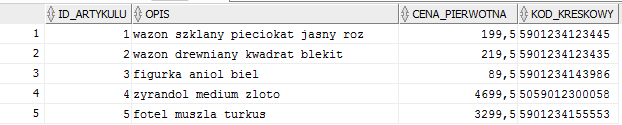
END LOOP;

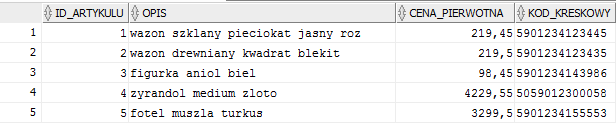
CLOSE artykuly;

END;

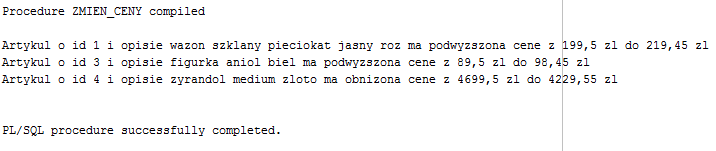
set SERVEROUTPUT ON;

exec zmien\_ceny(4000,200);

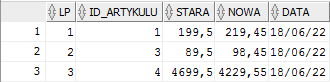
**Artykul przed**

**Artykuł po**

**Script output**



**Hist\_cen**



1. **Procedura bez Kursora**
2. Procedura sluzaca do wstawiania ogólnych danych dotyczących nowego zamówienia.Jako parametry podaje nr\_tel\_klienta i sposób\_realizacji Procedura powinna sprawdzic, czy podany nr tel klienta istnieje - (w przeciwnym przypadku zglaszamy blad). Procedura powinna równiez nadac unikalne id nowemu zamówieniu obliczone jako maksymalne id\_zamowienia w tabeli + 1. Data zamówienia to data dzisiejsza, opiekun zamówienia, to pracownik najmłodszy stazem w firmie, a dodatkowe koszty wynoszą tyle, co średnia z dotychczas realizowanych przesyłek, które mają ten sam sposób realizacji. jeśli dany sposób realizacji nie istnieje, to przypisujemy mu koszt póki co równy najmniejszemu ze wszystkich dotychczas zrealizowanych.

CREATE OR REPLACE PROCEDURE

dodaj\_dane\_zamowienia (nr\_tel\_klienta varchar2,

sposob\_realizacji\_zamowienia varchar2)

AS

spr\_numer number;

zweryfikowane\_id\_klienta number;

zweryfikowane\_id\_opiekuna number;

spr\_sposob\_realizacji number;

id\_zamowienia\_nowy Dane\_zamowienia.id\_zamowienia%type;

nr\_tel\_nie\_ma exception;

srednia\_wartosc\_kosztow Dane\_zamowienia.koszty\_dodatkowe%type;

BEGIN

select count(\*) into spr\_numer from Klient where nr\_telefonu=nr\_tel\_klienta;

IF spr\_numer > 0 THEN

SELECT id\_klienta into zweryfikowane\_id\_klienta FROM Klient WHERE

nr\_telefonu = nr\_tel\_klienta;

SELECT(\*) into spr\_sposob\_realizacji from Dane\_zamowienia WHERE

sposob\_realizacji=sposob\_realizacji\_zamowienia;

IF spr\_sposob\_realizacji > 0 THEN

SELECT AVG(koszty\_dodatkowe) into srednia\_wartosc\_kosztow FROM

Dane\_zamowienia WHERE

sposob\_realizacji=sposob\_realizacji\_zamowienia;

ELSE

SELECT MIN (koszty\_dodatkowe) into srednia\_wartosc\_kosztow

FROM Dane\_zamowienia;

END IF;

SELECT MAX(id\_zamowienia) into id\_zamowienia\_nowy from Dane\_zamowienia;

SELECT id\_opiekuna into zweryfikowane\_id\_opiekuna from Opiekun\_zamowienia

WHERE data\_zatrudnienia in (select max(data\_zatrudnienia)

FROM Opiekun\_zamowienia);

INSERT INTO Dane\_zamowienia VALUES

(id\_zamowienia\_nowy+1

,zweryfikowane\_id\_klienta

,zweryfikowane\_id\_opiekuna,sysdate,sposob\_realizacji\_zamowienia

,srednia\_wartosc\_kosztow);

COMMIT;

ELSE

RAISE nr\_tel\_nie\_ma;

END IF

EXCEPTION

WHEN nr\_tel\_nie\_ma THEN

dbMs\_OUTPUT.PUT\_LINE ('telefon ' || nr\_tel\_klienta || ' nie istnieje');

END;

SHOW ERRORS;

SET SERVEROUTPUT ON ;

EXEC dodaj\_dane\_zamowienia('+48509765345','przesylka kurierska DHL');

SHOW ERRORS;

SET SERVEROUTPUT ON ;

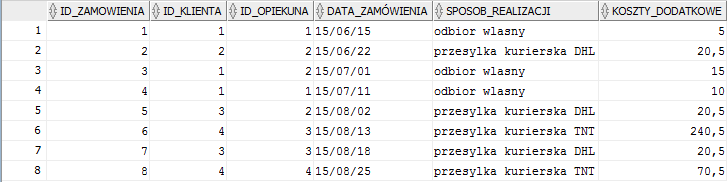
EXEC dodaj\_dane\_zamowienia('+48608735398','przesylka kurierska TNT');

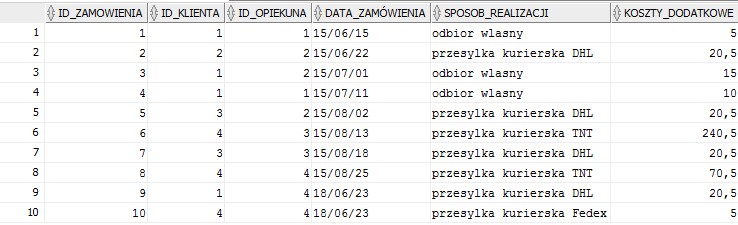
SHOW ERRORS;

SET SERVEROUTPUT ON ;

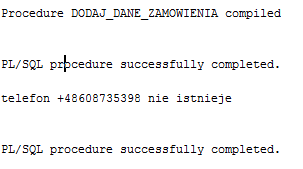
EXEC dodaj\_dane\_zamowienia('+4201764025','przesylka kurierska Fedex');

**Dane\_zamowienia przed**



**Dane\_zamowienia po**

**Script output**



1. **Wyzwalacz**
2. wyzwalacz, którzy nie pozwoli dodać do systemu artykulu z ceną wyższą niż 6000zł.

CREATE OR REPLACE TRIGGER sprawdz\_cene

BEFORE INSERT ON Artykul

FOR EACH ROW

BEGIN

IF (:NEW.cena\_pierwotna>6000) THEN

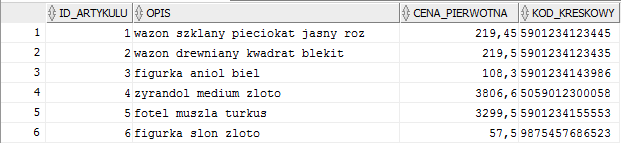
Raise\_application\_error(-20503, 'Cena za artykul nie moze byc wyzsza niz 6000zl');

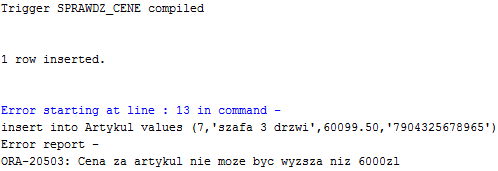
END IF;

END;

INSERT Artykul values (6,'figurka slon zloto',57.50,'9875457686523');

INSERT INTO Artykul values (7,'szafa 3 drzwi',60099.50,'7904325678965'**);**

**Artykuł – dodana wartość**

**Script output**

1. wyzwalacz, który nie pozwoli podnieść ceny o więcej, nić 15%.

CREATE OR REPLACE TRIGGER podnies\_cene

BEFORE UPDATE ON Artykul

FOR EACH ROW

BEGIN

IF (:NEW.cena\_pierwotna>(:OLD.cena\_pierwotna\*1.15))

THEN

Raise\_application\_error(-20505, 'Nie mozna wykonac, Dopuszczalna podwyzszka wynosi max. 15%');

END if;

END;

UPDATE Artykul SET cena\_pierwotna=110 where id\_artykulu=6;

UPDATE Artykul SET cena\_pierwotna=230 where id\_artykulu=2;

**Artykuł po update**

**Script output**