**Bazy danych - sprawozdanie z projektu**

# Podstawowe informacje

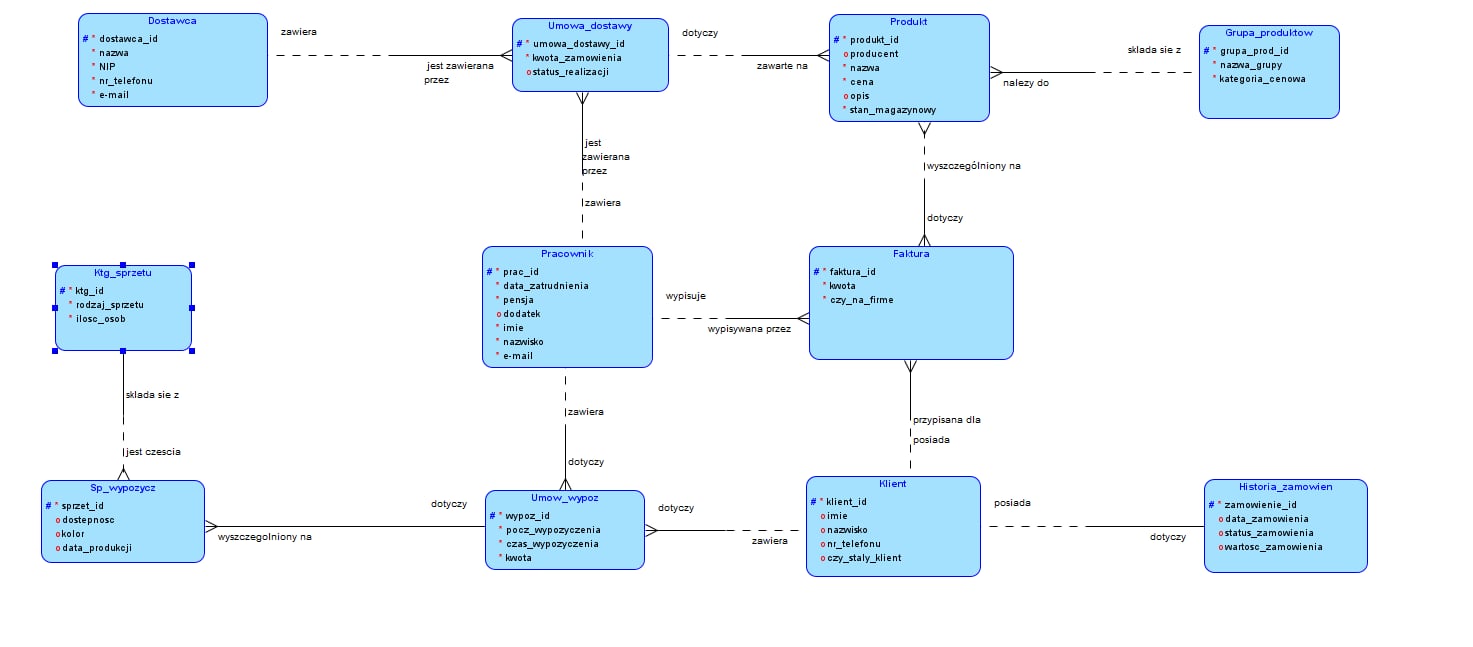
* **Autorzy projektu:** Hubert Dalmata 127752, Łukasz Kolanowski 127793
* **Nazwa grupy:** grupa 3
* **Temat:** Sklep wędkarski z wypożyczalnią sprzętu wodnego
* **Wybrany zakres (ocena):** 5
* **Baza danych: NAZWA:** prywatne2 **HASŁO:** LK88219LK

# Wprowadzenie – opis studium przypadku

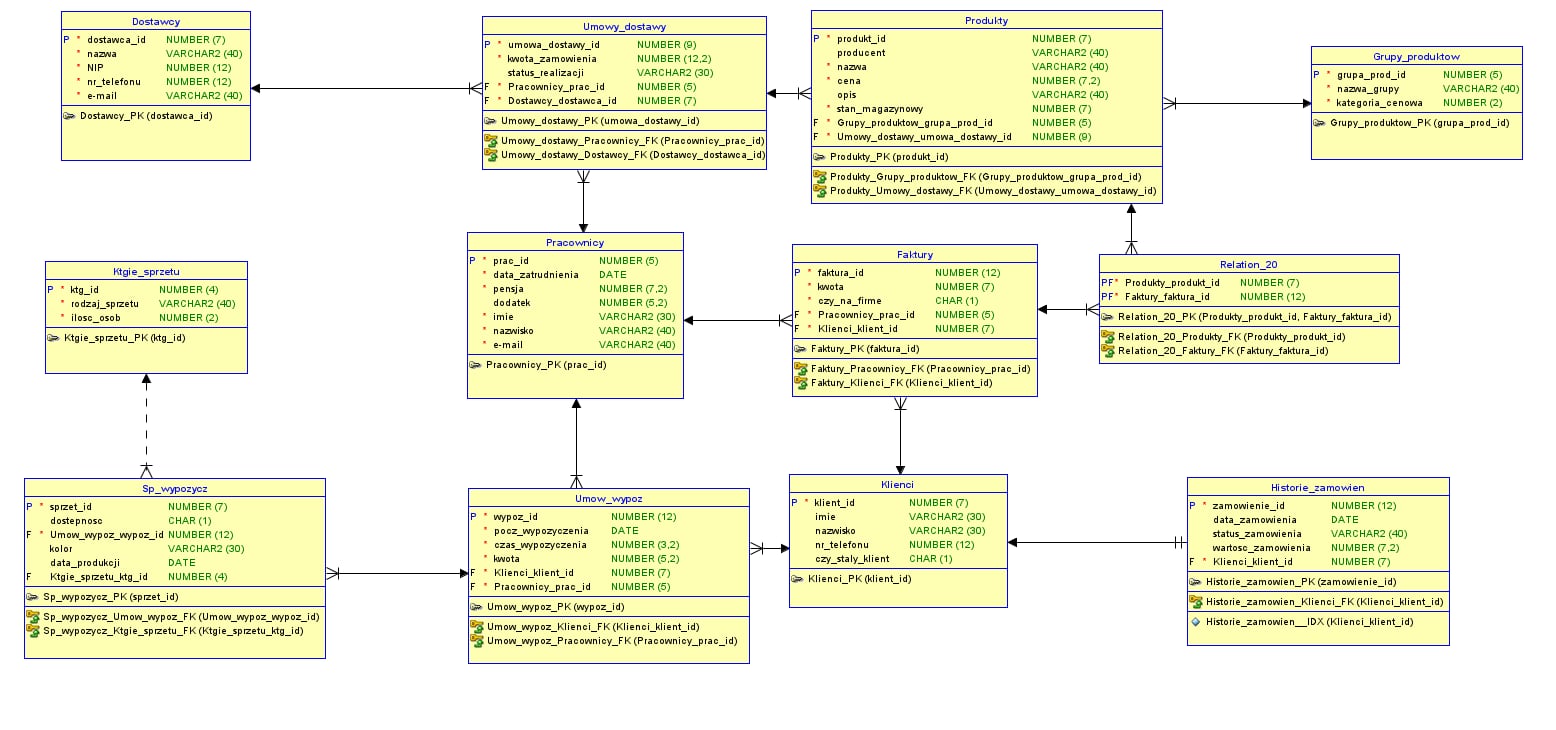
Krótkie rozwinięcie realizowanego tematu, przykładowe zastosowanie projektu itd.

Projekt obejmuje zaprojektowanie bazy danych, która obsługuje działalność fizycznego sklepu wędkarskiego z możliwością wypożyczenia sprzętu wodnego. W naszym projekcie zawarliśmy wszystkie kluczowe aspekty związane z prowadzeniem sklepu, takie jak: klienci, dostawcy, pracownicy sklepu, umowy i faktury, produkty oraz sprzęt do wypożyczenia. Naszą bazę danych można zastosować w obszarze analizy sprzedaży i funkcjonowania sklepu. Takie zagadnienia jak faktury zakupów i historia zakupowa klienta również są zawarte w projekcie. Informacje o pracownikach zawarte w bazie danych mogą nam posłużyć do mierzenia efektywności i wydajności zespołu. Na koniec warto wspomnieć, że nasze rozwiązanie pozwala sprawdzić dostępność sprzętu wodnego i monitorować czas wypożyczenia, oraz wyciągnąć informacje na temat umowy.

# Model związków encji



# Model relacyjnej bazy danych



# Schemat bazy - tabele

Opisy tabel: nazwy, polecenia DDL i DML, klucze podstawowe i klucze obce.

-- Generated by Oracle SQL Developer Data Modeler 24.3.0.240.1210

-- at: 2025-01-22 16:28:45 CET

-- site: Oracle Database 11g

-- type: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE dostawcy (

dostawca\_id NUMBER(7) NOT NULL,

nazwa VARCHAR2(40) NOT NULL,

nip NUMBER(12) NOT NULL,

nr\_telefonu NUMBER(12) NOT NULL,

"e-mail" VARCHAR2(40) NOT NULL

);

ALTER TABLE dostawcy ADD CONSTRAINT dostawcy\_pk PRIMARY KEY ( dostawca\_id );

CREATE TABLE faktury (

faktura\_id NUMBER(12) NOT NULL,

kwota NUMBER(7) NOT NULL,

czy\_na\_firme CHAR(1) NOT NULL,

pracownicy\_prac\_id NUMBER(5) NOT NULL,

klienci\_klient\_id NUMBER(7) NOT NULL

);

ALTER TABLE faktury ADD CONSTRAINT faktury\_pk PRIMARY KEY ( faktura\_id );

CREATE TABLE grupy\_produktow (

grupa\_prod\_id NUMBER(5) NOT NULL,

nazwa\_grupy VARCHAR2(40) NOT NULL,

kategoria\_cenowa NUMBER(2) NOT NULL

);

ALTER TABLE grupy\_produktow ADD CONSTRAINT grupy\_produktow\_pk PRIMARY KEY ( grupa\_prod\_id );

CREATE TABLE historie\_zamowien (

zamowienie\_id NUMBER(12) NOT NULL,

data\_zamowienia DATE,

status\_zamowienia VARCHAR2(40),

wartosc\_zamowienia NUMBER(7, 2),

klienci\_klient\_id NUMBER(7) NOT NULL

);

CREATE UNIQUE INDEX historie\_zamowien\_\_idx ON

historie\_zamowien (

klienci\_klient\_id

ASC );

ALTER TABLE historie\_zamowien ADD CONSTRAINT historie\_zamowien\_pk PRIMARY KEY ( zamowienie\_id );

CREATE TABLE klienci (

klient\_id NUMBER(7) NOT NULL,

imie VARCHAR2(30),

nazwisko VARCHAR2(30),

nr\_telefonu NUMBER(12),

czy\_staly\_klient CHAR(1)

);

ALTER TABLE klienci ADD CONSTRAINT klienci\_pk PRIMARY KEY ( klient\_id );

CREATE TABLE ktgie\_sprzetu (

ktg\_id NUMBER(4) NOT NULL,

rodzaj\_sprzetu VARCHAR2(40) NOT NULL,

ilosc\_osob NUMBER(2) NOT NULL

);

ALTER TABLE ktgie\_sprzetu ADD CONSTRAINT ktgie\_sprzetu\_pk PRIMARY KEY ( ktg\_id );

CREATE TABLE pracownicy (

prac\_id NUMBER(5) NOT NULL,

data\_zatrudnienia DATE NOT NULL,

pensja NUMBER(7, 2) NOT NULL,

dodatek NUMBER(5, 2),

imie VARCHAR2(30) NOT NULL,

nazwisko VARCHAR2(40) NOT NULL,

"e-mail" VARCHAR2(40) NOT NULL

);

ALTER TABLE pracownicy ADD CONSTRAINT pracownicy\_pk PRIMARY KEY ( prac\_id );

CREATE TABLE produkty (

produkt\_id NUMBER(7) NOT NULL,

producent VARCHAR2(40),

nazwa VARCHAR2(40) NOT NULL,

cena NUMBER(7, 2) NOT NULL,

opis VARCHAR2(40),

stan\_magazynowy NUMBER(7) NOT NULL,

grupy\_produktow\_grupa\_prod\_id NUMBER(5) NOT NULL,

umowy\_dostawy\_umowa\_dostawy\_id NUMBER(9) NOT NULL

);

ALTER TABLE produkty ADD CONSTRAINT produkty\_pk PRIMARY KEY ( produkt\_id );

CREATE TABLE relation\_20 (

produkty\_produkt\_id NUMBER(7) NOT NULL,

faktury\_faktura\_id NUMBER(12) NOT NULL

);

ALTER TABLE relation\_20 ADD CONSTRAINT relation\_20\_pk PRIMARY KEY ( produkty\_produkt\_id,

faktury\_faktura\_id );

CREATE TABLE sp\_wypozycz (

sprzet\_id NUMBER(7) NOT NULL,

dostepnosc CHAR(1),

kolor VARCHAR2(30),

data\_produkcji DATE,

umow\_wypoz\_wypoz\_id NUMBER(12) NOT NULL,

ktgie\_sprzetu\_ktg\_id NUMBER(4)

);

ALTER TABLE sp\_wypozycz ADD CONSTRAINT sp\_wypozycz\_pk PRIMARY KEY ( sprzet\_id );

CREATE TABLE umow\_wypoz (

wypoz\_id NUMBER(12) NOT NULL,

pocz\_wypozyczenia DATE NOT NULL,

czas\_wypozyczenia NUMBER(3, 2) NOT NULL,

kwota NUMBER(5, 2) NOT NULL,

klienci\_klient\_id NUMBER(7) NOT NULL,

pracownicy\_prac\_id NUMBER(5) NOT NULL

);

ALTER TABLE umow\_wypoz ADD CONSTRAINT umow\_wypoz\_pk PRIMARY KEY ( wypoz\_id );

CREATE TABLE umowy\_dostawy (

umowa\_dostawy\_id NUMBER(9) NOT NULL,

kwota\_zamowienia NUMBER(12, 2) NOT NULL,

status\_realizacji VARCHAR2(30),

pracownicy\_prac\_id NUMBER(5) NOT NULL,

dostawcy\_dostawca\_id NUMBER(7) NOT NULL

);

ALTER TABLE umowy\_dostawy ADD CONSTRAINT umowy\_dostawy\_pk PRIMARY KEY ( umowa\_dostawy\_id );

ALTER TABLE faktury

ADD CONSTRAINT faktury\_klienci\_fk FOREIGN KEY ( klienci\_klient\_id )

REFERENCES klienci ( klient\_id );

ALTER TABLE faktury

ADD CONSTRAINT faktury\_pracownicy\_fk FOREIGN KEY ( pracownicy\_prac\_id )

REFERENCES pracownicy ( prac\_id );

ALTER TABLE historie\_zamowien

ADD CONSTRAINT historie\_zamowien\_klienci\_fk FOREIGN KEY ( klienci\_klient\_id )

REFERENCES klienci ( klient\_id );

ALTER TABLE produkty

ADD CONSTRAINT produkty\_grupy\_produktow\_fk FOREIGN KEY ( grupy\_produktow\_grupa\_prod\_id )

REFERENCES grupy\_produktow ( grupa\_prod\_id );

ALTER TABLE produkty

ADD CONSTRAINT produkty\_umowy\_dostawy\_fk FOREIGN KEY ( umowy\_dostawy\_umowa\_dostawy\_id )

REFERENCES umowy\_dostawy ( umowa\_dostawy\_id );

ALTER TABLE relation\_20

ADD CONSTRAINT relation\_20\_faktury\_fk FOREIGN KEY ( faktury\_faktura\_id )

REFERENCES faktury ( faktura\_id );

ALTER TABLE relation\_20

ADD CONSTRAINT relation\_20\_produkty\_fk FOREIGN KEY ( produkty\_produkt\_id )

REFERENCES produkty ( produkt\_id );

ALTER TABLE sp\_wypozycz

ADD CONSTRAINT sp\_wypozycz\_ktgie\_sprzetu\_fk FOREIGN KEY ( ktgie\_sprzetu\_ktg\_id )

REFERENCES ktgie\_sprzetu ( ktg\_id );

ALTER TABLE sp\_wypozycz

ADD CONSTRAINT sp\_wypozycz\_umow\_wypoz\_fk FOREIGN KEY ( umow\_wypoz\_wypoz\_id )

REFERENCES umow\_wypoz ( wypoz\_id );

ALTER TABLE umow\_wypoz

ADD CONSTRAINT umow\_wypoz\_klienci\_fk FOREIGN KEY ( klienci\_klient\_id )

REFERENCES klienci ( klient\_id );

ALTER TABLE umow\_wypoz

ADD CONSTRAINT umow\_wypoz\_pracownicy\_fk FOREIGN KEY ( pracownicy\_prac\_id )

REFERENCES pracownicy ( prac\_id );

ALTER TABLE umowy\_dostawy

ADD CONSTRAINT umowy\_dostawy\_dostawcy\_fk FOREIGN KEY ( dostawcy\_dostawca\_id )

REFERENCES dostawcy ( dostawca\_id );

ALTER TABLE umowy\_dostawy

ADD CONSTRAINT umowy\_dostawy\_pracownicy\_fk FOREIGN KEY ( pracownicy\_prac\_id )

REFERENCES pracownicy ( prac\_id );

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 12

-- CREATE INDEX 1

-- ALTER TABLE 25

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 0

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

-- DROP TABLESPACE 0

-- DROP DATABASE 0

-- REDACTION POLICY 0

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

-- ERRORS 0

-- WARNINGS 0

--DOSTAWCY

INSERT INTO dostawcy (dostawca\_id, nazwa, nip, nr\_telefonu, "e-mail") VALUES

(15, 'RybyPol', 111111111111, 543987654, 'rybypol@rybypol.pl');

INSERT INTO dostawcy (dostawca\_id, nazwa, nip, nr\_telefonu, "e-mail") VALUES

(16, 'ZanetyPol', 222222222222, 685983853, 'kontakt@zanetypol.pl');

INSERT INTO dostawcy (dostawca\_id, nazwa, nip, nr\_telefonu, "e-mail") VALUES

(17, 'WszystkoDlaRybaka', 333333333333, 386947594, 'kontakt@wszystkodlarybaka.pl');

INSERT INTO dostawcy (dostawca\_id, nazwa, nip, nr\_telefonu, "e-mail") VALUES

(18, 'SupyPolskie', 444444444444, 490765239, 'kontakt@supypolskie.com');

INSERT INTO dostawcy (dostawca\_id, nazwa, nip, nr\_telefonu, "e-mail") VALUES

(19, 'WedkiDlaCiebie', 555555555, 358940594, 'biuro@wedkidlaciebie.pl');

--KLIENCI

INSERT INTO klienci (klient\_id, imie, nazwisko, nr\_telefonu, czy\_staly\_klient) VALUES

(seq\_10.NEXTVAL, 'Marek', 'Sandacz', 398458800, 'T');

INSERT INTO klienci (klient\_id, imie, nazwisko, nr\_telefonu, czy\_staly\_klient) VALUES

(seq\_10.NEXTVAL, 'Radoslaw', 'Karp', 397307428, 'T');

INSERT INTO klienci (klient\_id, imie, nazwisko, nr\_telefonu, czy\_staly\_klient) VALUES

(seq\_10.NEXTVAL, 'Euzebiusz', 'Miruna', 457790800, 'F');

INSERT INTO klienci (klient\_id, imie, nazwisko, nr\_telefonu, czy\_staly\_klient) VALUES

(seq\_10.NEXTVAL, 'Miroslaw', 'Karas', 118458840, 'T');

INSERT INTO klienci (klient\_id, imie, nazwisko, nr\_telefonu, czy\_staly\_klient) VALUES

(seq\_10.NEXTVAL, 'Tomasz', 'Wegorz', 318118830, 'F');

--PRACOWNICY

INSERT INTO pracownicy (prac\_id, data\_zatrudnienia, pensja, dodatek, imie, nazwisko, "e-mail") VALUES

(1, date '2010-04-08', 5000, 400, 'Andrzej', 'Sprzedawiecki', 'a.sprzedawiecki@wedkarztoty.pl');

INSERT INTO pracownicy (prac\_id, data\_zatrudnienia, pensja, dodatek, imie, nazwisko, "e-mail") VALUES

(2, date '2013-10-09', 7000, 900, 'Radoslaw', 'Wedkarski', 'r.wedkarski@wedkarztoty.pl');

INSERT INTO pracownicy (prac\_id, data\_zatrudnienia, pensja, dodatek, imie, nazwisko, "e-mail") VALUES

(3, date '2019-04-04', 12000, 250, 'Euzebiusz', 'Pracowniczy', 'e.pracowniczy@wedkarztoty.pl');

INSERT INTO pracownicy (prac\_id, data\_zatrudnienia, pensja, dodatek, imie, nazwisko, "e-mail") VALUES

(4, date '2020-07-12', 5000, NULL, 'Arkadiusz', 'Lodzianin', 'a.lodzianin@wedkarztoty.pl');

INSERT INTO pracownicy (prac\_id, data\_zatrudnienia, pensja, dodatek, imie, nazwisko, "e-mail") VALUES

(5, date '2021-03-11', 4600, NULL, 'Kuba', 'Jablkowski', 'k.jablkowski@wedkarztoty.pl');

--HISTORIE\_ZAMOWIEN

INSERT INTO historie\_zamowien (zamowienie\_id, data\_zamowienia, status\_zamowienia, wartosc\_zamowienia, klienci\_klient\_id) VALUES

(1, date '2016-01-01', 'Zrealizowane', 432.45, 10);

INSERT INTO historie\_zamowien (zamowienie\_id, data\_zamowienia, status\_zamowienia, wartosc\_zamowienia, klienci\_klient\_id) VALUES

(2, date '2019-07-08', 'Zrealizowane', 190.50, 20);

INSERT INTO historie\_zamowien (zamowienie\_id, data\_zamowienia, status\_zamowienia, wartosc\_zamowienia, klienci\_klient\_id) VALUES

(3, date '2022-10-03', 'Zrealizowane', 234.98, 30);

INSERT INTO historie\_zamowien (zamowienie\_id, data\_zamowienia, status\_zamowienia, wartosc\_zamowienia, klienci\_klient\_id) VALUES

(4, date '2023-03-02', 'Anulowane', 89.99, 40);

INSERT INTO historie\_zamowien (zamowienie\_id, data\_zamowienia, status\_zamowienia, wartosc\_zamowienia, klienci\_klient\_id) VALUES

(5, date '2024-11-10', 'W trakcie realizacji', 139.99, 50);

--FAKTURY

INSERT INTO faktury (faktura\_id, kwota, czy\_na\_firme, pracownicy\_prac\_id, klienci\_klient\_id) VALUES

(1, 7482, 'T', 1, 20);

INSERT INTO faktury (faktura\_id, kwota, czy\_na\_firme, pracownicy\_prac\_id, klienci\_klient\_id) VALUES

(2, 412, 'F', 3, 30);

INSERT INTO faktury (faktura\_id, kwota, czy\_na\_firme, pracownicy\_prac\_id, klienci\_klient\_id) VALUES

(3, 51, 'T', 2, 10);

INSERT INTO faktury (faktura\_id, kwota, czy\_na\_firme, pracownicy\_prac\_id, klienci\_klient\_id) VALUES

(4, 421, 'F', 4, 50);

INSERT INTO faktury (faktura\_id, kwota, czy\_na\_firme, pracownicy\_prac\_id, klienci\_klient\_id) VALUES

(5, 5675, 'T', 1, 30);

--GRUPY\_PRODUKTOW

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'wedki', 5);

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'zanety', 2);

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'haczyki', 1);

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'linki', 2);

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'podbieraki', 3);

INSERT INTO grupy\_produktow (grupa\_prod\_id, nazwa\_grupy, kategoria\_cenowa) VALUES

(seq\_1.NEXTVAL, 'kolowrotki', 4);

--KTGIE\_SPRZETU

INSERT INTO ktgie\_sprzetu (ktg\_id, rodzaj\_sprzetu, ilosc\_osob) VALUES

(100, 'lodka', 4);

INSERT INTO ktgie\_sprzetu (ktg\_id, rodzaj\_sprzetu, ilosc\_osob) VALUES

(200, 'kajak', 2);

INSERT INTO ktgie\_sprzetu (ktg\_id, rodzaj\_sprzetu, ilosc\_osob) VALUES

(300, 'rower wodny', 4);

INSERT INTO ktgie\_sprzetu (ktg\_id, rodzaj\_sprzetu, ilosc\_osob) VALUES

(400, 'kapok', 1);

INSERT INTO ktgie\_sprzetu (ktg\_id, rodzaj\_sprzetu, ilosc\_osob) VALUES

(500, 'deska sup', 1);

--UMOW\_WYPOZ

INSERT INTO umow\_wypoz (wypoz\_id, pocz\_wypozyczenia, czas\_wypozyczenia, kwota, klienci\_klient\_id, pracownicy\_prac\_id) VALUES

(1, date '2025-01-01', 2.50, 50, 10, 1);

INSERT INTO umow\_wypoz (wypoz\_id, pocz\_wypozyczenia, czas\_wypozyczenia, kwota, klienci\_klient\_id, pracownicy\_prac\_id) VALUES

(2, date '2025-01-04', 1, 20, 20, 2);

INSERT INTO umow\_wypoz (wypoz\_id, pocz\_wypozyczenia, czas\_wypozyczenia, kwota, klienci\_klient\_id, pracownicy\_prac\_id) VALUES

(3, date '2025-01-07', 8, 100, 30, 1);

INSERT INTO umow\_wypoz (wypoz\_id, pocz\_wypozyczenia, czas\_wypozyczenia, kwota, klienci\_klient\_id, pracownicy\_prac\_id) VALUES

(4, date '2025-01-09', 1.5, 60, 40, 5);

INSERT INTO umow\_wypoz (wypoz\_id, pocz\_wypozyczenia, czas\_wypozyczenia, kwota, klienci\_klient\_id, pracownicy\_prac\_id) VALUES

(5, date '2024-12-12', 1.5, 30, 50, 2);

--UMOWY\_DOSTAWY

INSERT INTO umowy\_dostawy (umowa\_dostawy\_id, kwota\_zamowienia, status\_realizacji, pracownicy\_prac\_id, dostawcy\_dostawca\_id) VALUES

(1, 400.45, 'Zrealizowane', 1, 15);

INSERT INTO umowy\_dostawy (umowa\_dostawy\_id, kwota\_zamowienia, status\_realizacji, pracownicy\_prac\_id, dostawcy\_dostawca\_id) VALUES

(2, 2000.50, 'Zrealizowane', 2, 16);

INSERT INTO umowy\_dostawy (umowa\_dostawy\_id, kwota\_zamowienia, status\_realizacji, pracownicy\_prac\_id, dostawcy\_dostawca\_id) VALUES

(3, 680.98, 'Zrealizowane', 3, 18);

INSERT INTO umowy\_dostawy (umowa\_dostawy\_id, kwota\_zamowienia, status\_realizacji, pracownicy\_prac\_id, dostawcy\_dostawca\_id) VALUES

(4, 500.99, 'W trakcie realizacji', 4, 17);

INSERT INTO umowy\_dostawy (umowa\_dostawy\_id, kwota\_zamowienia, status\_realizacji, pracownicy\_prac\_id, dostawcy\_dostawca\_id) VALUES

(5, 5900.99, 'W trakcie realizacji', 5, 19);

--PRODUKTY

INSERT INTO produkty (produkt\_id, producent, nazwa, cena, opis, stan\_magazynowy, grupy\_produktow\_grupa\_prod\_id, umowy\_dostawy\_umowa\_dostawy\_id) VALUES

(1, 'Jaxon', 'wedka super ', 200, 'dobry wybor na karpia', 15, 1, 1);

INSERT INTO produkty (produkt\_id, producent, nazwa, cena, opis, stan\_magazynowy, grupy\_produktow\_grupa\_prod\_id, umowy\_dostawy\_umowa\_dostawy\_id) VALUES

(2, 'Kamatsu', 'kolowrotek piorun', 250, 'wytrzymaly i w uczciwej cenie', 21, 6, 2);

INSERT INTO produkty (produkt\_id, producent, nazwa, cena, opis, stan\_magazynowy, grupy\_produktow\_grupa\_prod\_id, umowy\_dostawy\_umowa\_dostawy\_id) VALUES

(3, 'Konger', 'linka karpiowa 4 mm', 15, 'polecam - Hubert Dalmata', 33, 4, 3);

INSERT INTO produkty (produkt\_id, producent, nazwa, cena, opis, stan\_magazynowy, grupy\_produktow\_grupa\_prod\_id, umowy\_dostawy\_umowa\_dostawy\_id) VALUES

(4, 'Fisker', 'podbierak medium', 95, 'na dorodne sztuki', 50, 5, 4);

INSERT INTO produkty (produkt\_id, producent, nazwa, cena, opis, stan\_magazynowy, grupy\_produktow\_grupa\_prod\_id, umowy\_dostawy\_umowa\_dostawy\_id) VALUES

(5, 'Dam', 'podbierak mini', 45, 'poreczny', 15, 5, 5);

--SP\_WYPOZYCZ

INSERT INTO sp\_wypozycz (sprzet\_id, dostepnosc, kolor, data\_produkcji, umow\_wypoz\_wypoz\_id, ktgie\_sprzetu\_ktg\_id) VALUES

(12, 'T', 'czerwony', date '2009-08-01', 5, 400);

INSERT INTO sp\_wypozycz (sprzet\_id, dostepnosc, kolor, data\_produkcji, umow\_wypoz\_wypoz\_id, ktgie\_sprzetu\_ktg\_id) VALUES

(13, 'T', 'srebrny', date '2010-04-06', 4, 300);

INSERT INTO sp\_wypozycz (sprzet\_id, dostepnosc, kolor, data\_produkcji, umow\_wypoz\_wypoz\_id, ktgie\_sprzetu\_ktg\_id) VALUES

(14, 'T', 'czarny', date '2011-01-08', 3, 200);

INSERT INTO sp\_wypozycz (sprzet\_id, dostepnosc, kolor, data\_produkcji, umow\_wypoz\_wypoz\_id, ktgie\_sprzetu\_ktg\_id) VALUES

(15, 'T', 'zielony', date '2017-12-12', 2, 100);

INSERT INTO sp\_wypozycz (sprzet\_id, dostepnosc, kolor, data\_produkcji, umow\_wypoz\_wypoz\_id, ktgie\_sprzetu\_ktg\_id) VALUES

(16, 'T', 'morski', date '2018-04-08', 1, 500);

**--Dane powyżej są fikcyjne**

# Przypadki użycia bazy danych

Opis przypadków użycia bazy danych wraz z poleceniami SELECT.

--1. Uzyskanie lacznej wartosci zamowien dla danego dostawcy

SELECT

dostawcy.nazwa AS nazwa\_dostawcy,

SUM(umowy\_dostawy.kwota\_zamowienia) AS laczna\_kwota\_zamowien

FROM

umowy\_dostawy

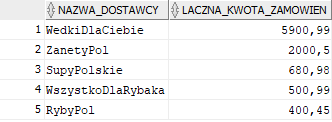
JOIN

dostawcy ON umowy\_dostawy.dostawcy\_dostawca\_id = dostawcy.dostawca\_id

GROUP BY

dostawcy.nazwa

ORDER BY

laczna\_kwota\_zamowien DESC;

--2. Uzyskanie danych pracowników, których średnia pensja jest wyższa niż średnia ogólna dla wszystkich pracowników.

SELECT \*

from pracownicy

where pensja+NVL(dodatek,0) > (SELECT AVG(pensja+ NVL(dodatek,0)) from pracownicy);



--3. Uzyskanie informacji, ktory pracownik, jakiemu klientowi wypożyczył sprzet w zwiazku z dana umowa, wraz z data produkcji sprzetu i dla ilu osob jest przeznaczony, uwzględniając tylko klientów, którzy mają przynajmniej jedno zrealizowane zamówienie.

SELECT

umow\_wypoz.wypoz\_id,

umow\_wypoz.pocz\_wypozyczenia,

klienci.imie || ' ' || klienci.nazwisko as klient,

pracownicy.imie || ' ' || pracownicy.nazwisko as pracownik,

TO\_CHAR(sp\_wypozycz.data\_produkcji, 'DD-Mon-YYYY',

'nls\_date\_language = Polish')as data\_produkcji\_sprzetu,

ktgie\_sprzetu.ilosc\_osob as ilu\_osobowy

FROM

umow\_wypoz

JOIN

klienci ON umow\_wypoz.klienci\_klient\_id = klienci.klient\_id

JOIN

pracownicy ON umow\_wypoz.pracownicy\_prac\_id = pracownicy.prac\_id

JOIN

sp\_wypozycz ON sp\_wypozycz.umow\_wypoz\_wypoz\_id = umow\_wypoz.wypoz\_id

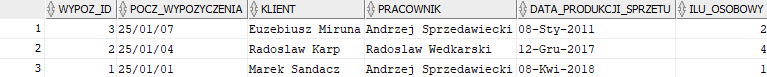
JOIN

ktgie\_sprzetu ON ktgie\_sprzetu.ktg\_id = sp\_wypozycz.ktgie\_sprzetu\_ktg\_id

WHERE

klienci.klient\_id IN (

SELECT klienci\_klient\_id FROM historie\_zamowien WHERE historie\_zamowien.status\_zamowienia ='Zrealizowane');



--4. Uzyskanie informacji ile czasu przepracowal pracownik, w dniach, miesiacach i latach

SELECT

imie,

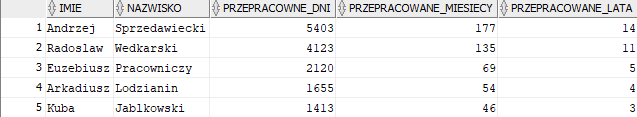
nazwisko,

FLOOR(SYSDATE -data\_zatrudnienia) AS przepracowne\_dni,

FLOOR(MONTHS\_BETWEEN(SYSDATE, data\_zatrudnienia)) AS przepracowane\_miesiecy,

FLOOR(MONTHS\_BETWEEN(SYSDATE, data\_zatrudnienia) / 12) AS przepracowane\_lata

FROM pracownicy;



--5. Uzyskanie liczby zamówień i łącznej wartości zamówień dla każdego klienta

SELECT

klienci.klient\_id,

SUBSTR(klienci.imie, 1,1) ||'. ' || klienci.nazwisko AS imie\_i\_nazwisko\_klienta,

COUNT(historie\_zamowien.zamowienie\_id) AS liczba\_zamowien,

SUM(historie\_zamowien.wartosc\_zamowienia) AS suma\_zamowien

FROM

klienci

JOIN

historie\_zamowien ON klienci.klient\_id = historie\_zamowien.klienci\_klient\_id

GROUP BY

klienci.klient\_id, klienci.imie, klienci.nazwisko

ORDER BY suma\_zamowien DESC;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

--6. Uzyskanie informacji o produktach drozszych niz wedka super wraz z grupa, do jakiej nalezy dany produkt

SELECT produkt\_id, producent, nazwa, grupy\_produktow.nazwa\_grupy

FROM produkty

JOIN grupy\_produktow ON grupy\_produktow.grupa\_prod\_id = produkty.grupy\_produktow\_grupa\_prod\_id

WHERE cena > (SELECT cena

FROM produkty

WHERE nazwa = 'wedka super ');



--7. Uzyskanie informacji o sprzetach dedykowanych dla co najmniej dwóch osób

SELECT s.sprzet\_id, s.kolor, k.ilosc\_osob

FROM sp\_wypozycz s

JOIN (SELECT ktg\_id, rodzaj\_sprzetu, ilosc\_osob FROM ktgie\_sprzetu WHERE ilosc\_osob >= 2) k

ON k.ktg\_id = s.ktgie\_sprzetu\_ktg\_id;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

--8. Uzyskanie informacji jacy dostawcy dostarczaja wedki

SELECT d.nazwa as Kto\_dostarcza\_wedki from dostawcy d

JOIN umowy\_dostawy um on um.dostawcy\_dostawca\_id=d.dostawca\_id

JOIN produkty p on p.UMOWY\_DOSTAWY\_UMOWA\_DOSTAWY\_ID=um.umowa\_dostawy\_id

WHERE INSTR(p.nazwa,'wedka')>0;



# Pozostałe obiekty bazy danych

**SEKWENCERY :**

Sekwencer poniżej służy do generowania id dla tabeli klienci (1,2,3…):

CREATE SEQUENCE seq\_1

INCREMENT BY 1

START WITH 1

MAXVALUE 5000

NOCACHE

NOCYCLE;

Sekwencer poniżej służy do generowania id dla tabeli grupy\_produktow(10,20,30…):

CREATE SEQUENCE seq\_10

INCREMENT BY 10

START WITH 10

MAXVALUE 5000

NOCACHE

NOCYCLE;

**PERSPEKTYWY:**

--1. Uzyskanie łącznej wartości zamówień dla danego dostawcy

CREATE VIEW v1

AS SELECT

dostawcy.nazwa AS nazwa\_dostawcy,

SUM(umowy\_dostawy.kwota\_zamowienia) AS laczna\_kwota\_zamowien

FROM

umowy\_dostawy

JOIN

dostawcy ON umowy\_dostawy.dostawcy\_dostawca\_id = dostawcy.dostawca\_id

GROUP BY

dostawcy.nazwa

ORDER BY

laczna\_kwota\_zamowien DESC

WITH READ ONLY ;

--2. Uzyskanie danych pracowników, których średnia pensja jest wyższa niż średnia ogólna dla wszystkich pracowników.

CREATE VIEW v2

AS SELECT \*

from pracownicy

where pensja+NVL(dodatek,0) > (SELECT AVG(pensja+ NVL(dodatek,0)) from pracownicy)

WITH READ ONLY;

--3. Uzyskanie informacji, który pracownik, jakiemu klientowi wypożyczył sprzęt w związku z dana umowa, wraz z data produkcji sprzętu i dla ilu osób jest przeznaczony, uwzględniając tylko klientów, którzy mają przynajmniej jedno zrealizowane zamówienie.

CREATE VIEW v3

AS SELECT

umow\_wypoz.wypoz\_id,

umow\_wypoz.pocz\_wypozyczenia,

klienci.imie || ' ' || klienci.nazwisko as klient,

pracownicy.imie || ' ' || pracownicy.nazwisko as pracownik,

TO\_CHAR(sp\_wypozycz.data\_produkcji, 'DD-Mon-YYYY',

'nls\_date\_language = Polish')as data\_produkcji\_sprzetu,

ktgie\_sprzetu.ilosc\_osob as ilu\_osobowy

FROM

umow\_wypoz

JOIN

klienci ON umow\_wypoz.klienci\_klient\_id = klienci.klient\_id

JOIN

pracownicy ON umow\_wypoz.pracownicy\_prac\_id = pracownicy.prac\_id

JOIN

sp\_wypozycz ON sp\_wypozycz.umow\_wypoz\_wypoz\_id = umow\_wypoz.wypoz\_id

JOIN

ktgie\_sprzetu ON ktgie\_sprzetu.ktg\_id = sp\_wypozycz.ktgie\_sprzetu\_ktg\_id

WHERE

klienci.klient\_id IN (

SELECT klienci\_klient\_id FROM historie\_zamowien WHERE historie\_zamowien.status\_zamowienia ='Zrealizowane'

)

WITH READ ONLY;

--4. Uzyskanie informacji ile czasu przepracował pracownik, w dniach, miesiącach i latach

CREATE VIEW v4

AS SELECT

imie,

nazwisko,

FLOOR(SYSDATE -data\_zatrudnienia) AS przepracowne\_dni,

FLOOR(MONTHS\_BETWEEN(SYSDATE, data\_zatrudnienia)) AS przepracowane\_miesiecy,

FLOOR(MONTHS\_BETWEEN(SYSDATE, data\_zatrudnienia) / 12) AS przepracowane\_lata

FROM pracownicy

WITH READ ONLY;

--5. Uzyskanie liczby zamówień i łącznej wartości zamówień dla każdego klienta

CREATE VIEW v5

AS SELECT

klienci.klient\_id,

SUBSTR(klienci.imie, 1,1) ||'. ' || klienci.nazwisko AS imie\_i\_nazwisko\_klienta,

COUNT(historie\_zamowien.zamowienie\_id) AS liczba\_zamowien,

SUM(historie\_zamowien.wartosc\_zamowienia) AS suma\_zamowien

FROM

klienci

JOIN

historie\_zamowien ON klienci.klient\_id = historie\_zamowien.klienci\_klient\_id

GROUP BY

klienci.klient\_id, klienci.imie, klienci.nazwisko

ORDER BY

suma\_zamowien DESC

WITH READ ONLY;

--6. Uzyskanie informacji o produktach drozszych niz wedka super wraz z grupa, do jakiej nalezy dany produkt

CREATE VIEW v6

AS SELECT produkt\_id, producent, nazwa, grupy\_produktow.nazwa\_grupy

FROM produkty

JOIN grupy\_produktow ON grupy\_produktow.grupa\_prod\_id = produkty.grupy\_produktow\_grupa\_prod\_id

WHERE cena > (SELECT cena

FROM produkty

WHERE nazwa = 'wedka super ')

WITH READ ONLY;

--7. Uzyskanie informacji o sprzętach dedykowanych dla co najmniej dwóch osób

CREATE VIEW v7

AS SELECT s.sprzet\_id, s.kolor, k.ilosc\_osob

FROM sp\_wypozycz s

JOIN (SELECT ktg\_id, rodzaj\_sprzetu, ilosc\_osob FROM ktgie\_sprzetu WHERE ilosc\_osob >= 2) k

ON k.ktg\_id = s.ktgie\_sprzetu\_ktg\_id

WITH READ ONLY;

--8. Uzyskanie informacji jacy dostawcy dostarczają wędki

CREATE VIEW v8

AS SELECT d.nazwa as Kto\_dostarcza\_wedki from dostawcy d

JOIN umowy\_dostawy um on um.dostawcy\_dostawca\_id=d.dostawca\_id

JOIN produkty p on p.UMOWY\_DOSTAWY\_UMOWA\_DOSTAWY\_ID=um.umowa\_dostawy\_id

WHERE INSTR(p.nazwa,'wedka')>0

WITH READ ONLY;