--1. Podaj liczbę produktów o cenach mniejszych niż 10$ lubwiększych niż 20$

SELECT COUNT(UnitPrice)

FROM Products

WHERE UnitPrice BETWEEN 10 AND 20

--2. Podaj maksymalną cenę produktu dla produktów o cenachponiżej 20$

SELECT MAX(UnitPrice)

FROM Products

WHERE UnitPrice < 20

--3. Podaj maksymalną, minimalną i średnią cenę produktu dlproduktów sprzedawanych w butelkach (‘bottle’)

SELECT MAX(UnitPrice) as MAX ,MIN(UnitPrice) as MIN ,AVG(UnitPrice) as AVG

FROM Products

WHERE QuantityPerUnit LIKE '%bottle%'

--4. Wypisz informację o wszystkich produktach o ceniepowyżej średniej

SELECT \*

FROM Products

WHERE UnitPrice >(SELECT AVG(UnitPrice) FROM Products)

--5. Podaj wartość zamówienia o numerze 10250

SELECT SUM(UnitPrice\*Quantity \* (1-Discount))

FROM [Order Details]

WHERE OrderID = '10250'

--1. Podaj maksymalną cenę zamawianego produktu dlakażdego zamówienia. Posortuj zamówienia wgmaksymalnej ceny produktu

SELECT MAX(UnitPrice)

FROM [Order Details]

GROUP BY OrderID

ORDER BY 1 DESC

--2. Podaj maksymalną i minimalną cenę zamawianegoproduktu dla każdego zamówienia

SELECT MAX(UnitPrice) as MAX, MIN(UnitPrice) as MIN

FROM [Order Details]

GROUP BY OrderID

--3. Podaj liczbę zamówień dostarczanych przezposzczególnych spedytorów

SELECT COUNT(ShipVia), ShipVia

FROM Orders

GROUP BY ShipVia

--4. Który ze spedytorów był najaktywniejszy w 1997 roku?

SELECT TOP 1 ShipVia

FROM Orders

GROUP BY ShipVia

Order BY COUNT(ShipVia) DESC

--1. Wyświetl zamówienia dla których liczba pozycji zamówienia jest większa niż 5

SELECT OrderID

FROM [Order Details]

GROUP BY OrderID

HAVING COUNT(OrderID)>5

--2. Wyświetl klientów, dla których w 1998 roku zrealizowano więcej niż 8 zamówień (wyniki posortuj malejąco wg

--łącznej kwoty za dostarczenie zamówień dla każdego zklientów)

SELECT CustomerID

FROM Orders

WHERE YEAR(OrderDate) = 1998

Group BY CustomerID

HAVING COUNT(CustomerID) > 8

--1.Napisz polecenie, które wyświetla listę dzieci będących członkami biblioteki. Interesuje nas imię,

--nazwisko i data urodzenia dziecka.

SELECT firstname, lastname, birth\_date

FROM Juvenile

INNER JOIN Member

ON Juvenile.member\_no=member.member\_no

\_--2. Napisz polecenie, które podaje tytuły aktualnie wypożyczonych książek

SELECT title

FROM title

INNER JOIN loan

ON loan.title\_no = title.title\_no

\_--3. Podaj informacje o karach zapłaconych za przetrzymywanie książki o tytule ‘Tao Teh King’.

--Interesuje nas data oddania książki, ile dni była przetrzymywana i jaką zapłacono karę

SELECT in\_date, fine\_paid,day(due\_date)- day(in\_date) AS holded

FROM loanhist

INNER JOIN title

ON title.title\_no=loanhist.title\_no

WHERE title.title= 'Tao Teh King'

\_ --4.Napisz polecenie które podaje listę książek (numery ISBN)

--zarezerwowanych przez osobę o nazwisku: Stephen A. Graff

SELECT Isbn

FROM reservation

INNER JOIN member

ON member.member\_no=reservation.member\_no

WHERE member.lastname= 'Graff' AND member.firstname='Stephen' AND member.middleinitial = 'A'

--5.Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20 a 30, dla każdego produktu podaj dane adresowe dostawcy

SELECT ProductName, UnitPrice, Address, City

FROM Products

INNER JOIN Suppliers

ON Products.SupplierID=Suppliers.SupplierID

WHERE UnitPrice BETWEEN 20 AND 30

\_--6. Wybierz nazwy produktów oraz informacje o stanie magazynu dla produktów dostarczanych przez firmę ‘Tokyo Traders’

SELECT ProductName, UnitsInStock

FROM Products

INNER JOIN Suppliers

ON Suppliers.SupplierID = Products.SupplierID

WHERE Suppliers.CompanyName = 'Tokyo Traders'

\_--7. Czy są jacyś klienci którzy nie złożyli żadnego zamówienia w 1997 roku, jeśli tak to pokaż ich dane adresowe

SELECT Customers.CustomerID

FROM Customers

LEFT JOIN Orders

ON Orders.CustomerID=Customers.CustomerID

WHERE OrderDate IS NULL

\_--8. Wybierz nazwy i numery telefonów dostawców, dostarczających produkty, których aktualnie nie ma w magazynie

SELECT CompanyName, Phone

FROM Suppliers

INNER JOIN Products

ON Suppliers.SupplierID = Products.SupplierID

WHERE UnitsInStock = 0

--Wybierz nazwy i ceny produktów (baza northwind) o cenie jednostkowej pomiędzy 20 a 30, dla każdego produktu

--podaj dane adresowe dostawcy, interesują nas tylko produkty z kategorii ‘Meat/Poultry’

SELECT Products.ProductName, Products.UnitPrice, Suppliers.Address, Suppliers.City

FROM Products

INNER JOIN Suppliers

ON Products.ProductID = Suppliers.SupplierID

INNER JOIN Categories

ON Products.CategoryID=Categories.CategoryID

WHERE CategoryNAme = 'Meat/Poultry'

\_ --Wybierz nazwy i ceny produktów z kategorii ‘Confections’dla każdego produktu podaj nazwę dostawcy.

SELECT ProductName, UnitPrice, CompanyName

FROM Products

INNER JOIN Suppliers

ON Products.SupplierID = Suppliers.SupplierID

INNER JOIN Categories

ON Categories.CategoryID=Products.CategoryID

WHERE CategoryName = 'Confections'

\_ --Wybierz nazwy i numery telefonów klientów , którym w1997 roku przesyłki dostarczała firma ‘United Package’

SELECT Customers.CompanyName, Customers.Phone

FROM Customers

INNER JOIN Orders

ON Orders.CustomerID=Customers.CustomerID

INNER JOIN Shippers

ON Orders.ShipVia=Shippers.ShipperID

WHERE YEAR(ShippedDate) = '1997' AND Shippers.COmpanyName = 'United Package'

\_-- Wybierz nazwy i numery telefonów klientów, którzy kupowali produkty z kategorii ‘Confections’

SELECT Customers.Companyname, Customers.Phone

FROM Customers

INNER JOIN Orders

ON Orders.CustomerID=Customers.CustomerID

INNER JOIN [Order Details]

ON Orders.OrderID=[Order Details].OrderID

INNER JOIN Products

ON [Order Details].ProductID=Products.ProductID

INNER JOIN Categories

ON Products.CategoryID=Categories.CategoryID

WHERE CategoryName = 'Confections'

\_--1. Napisz polecenie, które wyświetla listę dzieci będących członkami biblioteki. Interesuje nas imię, nazwisko,

-- data urodzenia dziecka i adres zamieszkania dziecka.

SELECT member.firstname, member.lastname, juvenile.birth\_date, adult.street, adult.city, adult.state

FROM juvenile

INNER JOIN member

ON juvenile.member\_no=member.member\_no

INNER JOIN adult

ON juvenile.adult\_member\_no = adult.member\_no

--\_2.Napisz polecenie, które wyświetla listę dzieci będących członkami biblioteki. Interesuje nas imię, nazwisko,

-- data urodzenia dziecka, adres zamieszkania dziecka oraz imię i nazwisko rodzica.

SELECT childmember.firstname, childmember.lastname, juvenile.birth\_date, adult.street, adult.city, adult.state, adultmember.firstname, adultmember.lastname

FROM juvenile

INNER JOIN member AS childmember

ON juvenile.member\_no=childmember.member\_no

INNER JOIN adult

ON juvenile.adult\_member\_no = adult.member\_no

INNER JOIN member AS adultmember

ON adult.member\_no = adultmember.member\_no

\_--1. Napisz polecenie, które wyświetla pracowników oraz ich podwładnych.

SELECT A.FirstName + ' ' + A.LastName AS Pracownik , B.FirstName + ' ' + B.LastName AS Podwładny

FROM Employees AS A

INNER JOIN Employees as B

ON A.ReportsTo = B.EmployeeID

--\_2. Napisz polecenie, które wyświetla pracowników, którzy nie mają podwładnych.

SELECT B.FirstName + ' ' + B.LastName

FROM Employees AS A

RIGHT JOIN Employees as B

ON A.ReportsTo = B.EmployeeID

WHERE A.EmployeeID IS NULL

--1.Napisz polecenie, które wyświetla adresy członków biblioteki, którzy mają dzieci urodzone przed 1 stycznia 1996

SELECT street, city, state, zip

FROM adult

JOIN juvenile

ON juvenile.adult\_member\_no= adult.member\_no

WHERE year(birth\_date) <= 1995

--2.\_ Napisz polecenie, które wyświetla adresy członków biblioteki, którzy mają dzieci urodzone przed 1 stycznia

--1996. Interesują nas tylko adresy takich członków biblioteki, którzy aktualnie nie przetrzymują książek.

SELECT street, city, state, zip

FROM adult

JOIN juvenile

ON juvenile.adult\_member\_no= adult.member\_no

JOIN member

ON member.member\_no=adult.member\_no

LEFT JOIN loan

ON member.member\_no=loan.member\_no

WHERE year(birth\_date) <= 1995 AND loan.due\_date > GETDATE()

\_ --1.Napisz polecenie które zwraca imię i nazwisko (jako pojedynczą kolumnę – name), oraz informacje o adresie:

--ulica, miasto, stan kod (jako pojedynczą kolumnę – address) dla wszystkich dorosłych członków biblioteki

SELECT firstname+ ' '+ lastname AS name, street + city + state + zip AS address

FROM adult

JOIN member

on adult.member\_no=member.member\_no

\_-- 2.Napisz polecenie które zwraca informację o użytkownikach biblioteki o nr 250, 342, i 1675 (nr, imię i nazwisko

-- członka biblioteki) oraz informacje o zarezerwowanych książkach (isbn, data)

SELECt member.member\_no, firstname, lastname, isbn, log\_date

FROM member

LEFT JOIN reservation

ON reservation.member\_no = member.member\_no

WHERE member.member\_no = 250 OR member.member\_no = 342 OR member.member\_no = 1675

\_-- 3.Podaj listę członków biblioteki mieszkających w Arizonie (AZ), którzy mają więcej niż dwoje dzieci zapisanych

SELECT member.member\_no

FROM member

JOIN adult

ON adult.member\_no=member.member\_no

JOIN juvenile

ON juvenile.adult\_member\_no=adult.member\_no

WHERE adult.state = 'AZ'

group by member.member\_no

HAVING count(member.member\_no) >2

\_-- 4.Podaj listę członków biblioteki mieszkających w Arizonie (AZ) którzy mają więcej niż dwoje dzieci zapisanych

-- oraz takich którzy mieszkają w Kaliforni i mają więcej niż troje dzieci zapisanych do biblioteki

SELECT member.member\_no

FROM member

JOIN adult

ON adult.member\_no=member.member\_no

JOIN juvenile

ON juvenile.adult\_member\_no=adult.member\_no

WHERE adult.state = 'AZ'

group by member.member\_no

HAVING count(member.member\_no) >2

UNION

SELECT member.member\_no

FROM member

JOIN adult

ON adult.member\_no=member.member\_no

JOIN juvenile

ON juvenile.adult\_member\_no=adult.member\_no

WHERE adult.state = 'CA'

group by member.member\_no

HAVING count(member.member\_no) >3

--1. Dla każdej kategorii produktu (nazwa), podaj łączną liczbę zamówionych przez klienta jednostek.

SELECT CategoryName, count(Orders.CustomerID), Orders.CustomerID

FROM Categories

JOIN Products

ON Products.CategoryID=Categories.CategoryID

JOIN [Order Details]

ON [Order Details].ProductID=Products.ProductID

JOIN Orders

ON Orders.OrderID = [Order Details].OrderID

group by CategoryName, Orders.CustomerID

--2. Dla każdego zamówienia podaj łączną liczbę zamówionych jednostek oraz nazwę klienta.

SELECT CategoryName, count(Customers.CompanyName), Customers.CompanyName

FROM Categories

JOIN Products

ON Products.CategoryID=Categories.CategoryID

JOIN [Order Details]

ON [Order Details].ProductID=Products.ProductID

JOIN Orders

ON Orders.OrderID = [Order Details].OrderID

JOIN Customers

ON Orders.CustomerID=Customers.CustomerID

group by CategoryName, Customers.CompanyName

--3. Zmodyfikuj poprzedni przykład, aby pokazać tylko takie zamówienia, dla których łączna liczba jednostek

-- jest większa niż 250.

SELECT CategoryName, count(Customers.CompanyName), Customers.CompanyName

FROM Categories

JOIN Products

ON Products.CategoryID=Categories.CategoryID

JOIN [Order Details]

ON [Order Details].ProductID=Products.ProductID

JOIN Orders

ON Orders.OrderID = [Order Details].OrderID

JOIN Customers

ON Orders.CustomerID=Customers.CustomerID

group by CategoryName, Customers.CompanyName

HAVING count(Customers.CompanyName)>20

--4. Dla każdego klienta (nazwa) podaj nazwy towarów, które zamówił

SELECT CompanyName, ProductName

FROM Customers

Join Orders

ON Orders.CustomerID = Customers.CustomerID

JOIN [Order Details]

ON [Order Details].OrderID=Orders.OrderID

JOIN Products

ON Products.ProductID = [Order Details].ProductID

order by CompanyName

--5. Dla każdego klienta (nazwa) podaj wartość poszczególnych zamówień. Gdy klient nic nie zamówił też powinna

-- pojawić się informacja.

SELECT COmpanyName, SUM(UnitPrice\*Quantity\*(1-Discount)) AS sum

FROM Customers

LEFT Join Orders

ON Orders.CustomerID = Customers.CustomerID

JOIN [Order Details]

ON [Order Details].OrderID=Orders.OrderID

group by CompanyName

--6. Podaj czytelników (imię, nazwisko), którzy nigdy nie pożyczyli żadnej książki.

SELECT firstname,lastname

from member

LEFT JOIN loan

on loan.member\_no=member.member\_no

JOIN copy

ON loan.isbn=copy.isbn

JOIN loanhist

ON copy.isbn=loanhist.isbn

WHERE (loan.isbn IS NULL) AND (loanhist.isbn IS NULL)