

SQL I

Wybieranie danych z pojedynczej tabeli

Baza NORTHWIND

1. Wybierz nazwy i adresy wszystkich klientów

```
select CompanyName, Address from Customers
```

2. Wybierz nazwiska i numery telefonów pracowników

```
select LastName, HomePhone from Employees
```

3. Wybierz nazwy i ceny produktów

```
select ProductName, UnitPrice from Products
```

4. Pokaż wszystkie kategorie produktów (nazwy i opisy)

```
select CategoryName, Description  
from Categories
```

5. Pokaż nazwy i adresy stron www dostawców

```
select CompanyName, HomePage  
from Suppliers
```

6. Wybierz nazwy i adresy wszystkich klientów mających siedziby w Londynie

```
select CompanyName, Address from Customers  
where City = 'London'
```

```
select CompanyName, Address from Customers  
where City LIKE 'London'
```

7. Wybierz nazwy i adresy wszystkich klientów mających siedziby we Francji lub w Hiszpanii

```
select CompanyName, Address  
from Customers  
where Country LIKE 'Spain' or Country LIKE 'France'
```

8. Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20.00 a 30.00

```
select ProductName, UnitPrice  
from Products  
where UnitPrice between 20.00 and 30.00
```

9. Wybierz nazwy i ceny produktów z kategorii 'seafood'

```
select Products.ProductName, Products.UnitPrice  
from Products  
join Categories on Categories.CategoryID = Products.CategoryID  
where Categories.CategoryName like 'Seafood'
```

10. Wybierz nazwy produktów oraz inf. o stanie magazynu dla produktów dostarczanych przez firmę Tokyo Traders.

```
select SupplierID, CompanyName from Suppliers  
where CompanyName = 'Tokyo Traders'  
select ProductName, UnitsInStock from Products  
where SupplierID = 4
```

11. Wybierz nazwy produktów których nie ma w magazynie

```
select ProductName, UnitsInStock from Products
where UnitsInStock = 0
```

12. Szukamy informacji o produktach sprzedawanych w butelkach (bottle)

```
select * from Products
where QuantityPerUnit LIKE '%bottle%'
```

13. Wyszukaj informacje o stanowisku pracowników, których nazwiska zaczynają się na literę z zakresu od B do L

```
select * from Employees
where LastName LIKE '[B-L]%'
```

14. Wyszukaj informacje o stanowisku pracowników, których nazwiska zaczynają się na literę B lub L

```
select * from Employees
where LastName LIKE 'B%' OR LastName LIKE 'L%'
```

15. Znajdź nazwy kategorii, które w opisie zawierają przecinek

```
select CategoryName from Categories
where Description LIKE '%,%'
```

16. Znajdź klientów, którzy w swojej nazwie mają w którymś miejscu słowo .Store.

```
select CompanyName from Customers
where CompanyName LIKE '%Store%'
```

17. Szukamy informacji o produktach o cenach mniejszych niż 10 lub większych niż 20

```
select ProductName, UnitPrice from Products
where UnitPrice NOT BETWEEN 10 AND 20
```

```
select ProductName, UnitPrice from Products
where (UnitPrice < 10) OR (UnitPrice > 20)
```

18. Napisz instrukcję select tak aby wybrać numer zlecenia, datę zamówienia, numer klienta dla wszystkich niezrealizowanych jeszcze zleceń, dla których krajem odbiorcy jest Argentyna

```
select OrderID, CustomerID, OrderDate from Orders
where (ShippedDate > GETDATE() or ShippedDate is NULL) AND (ShipCountry = 'Argentina')
```

The GETDATE() function returns the current date and time from the SQL Server:

19. Wybierz nazwy i kraje wszystkich klientów, wyniki posortuj według kraju, w ramach danego kraju nazwy firm posortuj alfabetycznie

```
select CompanyName, Country from Customers
order by Country, CompanyName
```

20. Wybierz informacje o produktach (grupa, nazwa, cena), produkty posortuj wg grup a w grupach malejąco wg ceny

```
select ProductID, ProductName, CategoryID, UnitPrice from Products
order by CategoryID, UnitPrice desc
```

```
select ProductID, ProductName, CategoryID, UnitPrice from Products
order by 3, 4 desc
```

21. Wybierz nazwy i kraje wszystkich klientów mających siedziby w Japonii (Japan) lub we Włoszech (Italy), wyniki posortuj według kraju, w ramach danego kraju nazwy firm posortuj alfabetycznie

```
select CompanyName, Country from Customers
where Country in ('Italy', 'Japan')
order by Country, CompanyName
```

Wybieranie danych z pojedynczej tabeli

Baza LIBRARY

1. Napisz polecenie select, za pomocą którego uzyskasz tytuł i numer książki

```
SELECT title, title_no
FROM title
```

2. Napisz polecenie, które wybiera tytuł o numerze 10

```
SELECT title
FROM title
WHERE title_no = 10
```

3. Napisz polecenie, które wybiera numer czytelnika i karę dla tych czytelników, którzy mają kary między \$8.00 a \$9.00 (z tablicy loanhist)

```
SELECT DISTINCT member_no, fine_assessed
FROM loanhist
WHERE (fine_assessed BETWEEN 8.00 AND 9.00) --AND ((fine_paid is null) OR
(fine_assessed > fine_paid ))
```

4. Napisz polecenie select, za pomocą którego uzyskasz numer książki i autora z tablicy title dla wszystkich książek, których autorem jest Charles Dickens lub Jane Austen

```
SELECT title_no, author
FROM title
WHERE author = 'Jane Austen' OR author = 'Charles Dickens'
```

5. Napisz polecenie, które wybiera numer tytułu i tytuł dla wszystkich rekordów zawierających string .adventures. gdzieś w tytule.

```
SELECT title_no, title
FROM title
WHERE title LIKE '%adventures%'
```

6. Napisz polecenie, które wybiera numer czytelnika, karę oraz zapłaconą karę dla wszystkich, którzy jeszcze nie zapłacili.

```
SELECT member_no, fine_assessed, fine_paid
FROM loanhist
WHERE (fine_assessed > isnull(fine_paid, 0))
```

7. Napisz polecenie, które wybiera wszystkie unikalne pary miast i stanów z tablicy adult.

```
SELECT DISTINCT city, state
FROM adult
order by city
```

8. Napisz polecenie, które wybiera wszystkie tytuły z tablicy title i wyświetla je w porządku alfabetycznym.

```
SELECT title
FROM title
order by title
```

9. Napisz polecenie, które:

- wybiera numer członka biblioteki (member_no), isbn książki (isbn) i wartość naliczonej kary (fine_assessed) z tablicy loanhist dla wszystkich wypożyczeń dla których naliczono karę (wartość nie NULL w kolumnie fine_assessed)

```
SELECT member_no, isbn, fine_assessed
FROM loanhist
where (fine_assessed is not NULL) AND (fine_assessed > 0)
order by fine_assessed desc
```

- stwórz kolumnę wyliczeniową zawierającą podwojoną wartość kolumny fine_assessed

- stwórz alias .double fine. dla tej kolumny

```
SELECT member_no, isbn, fine_assessed, (2*fine_assessed) AS 'double fine'
FROM loanhist
where (fine_assessed is not NULL) AND (fine_assessed > 0)
```

`order by fine_assessed desc`

10. Napisz polecenie, które

- generuje pojedynczą kolumnę, która zawiera kolumny: `firstname` (imię członka biblioteki), `middleinitial` (inicjał drugiego imienia) i `lastname` (nazwisko) z tablicy `member` dla wszystkich członków biblioteki, którzy nazywają się Anderson
- nazwij tak powstałą kolumnę `email_name` (użyj aliasu `email_name` dla kolumny)

```
SELECT firstname+middleinitial+lastname AS email_name
FROM member
WHERE lastname LIKE 'Anderson'
```

- zmodyfikuj polecenie, tak by zwróciło listę proponowanych loginów e-mail. utworzonych przez połączenie imienia członka biblioteki, z inicjałem drugiego imienia i pierwszymi dwoma literami nazwiska (wszystko małymi literami).
 - Wykorzystaj funkcję `SUBSTRING` do uzyskania części kolumny znakowej oraz `LOWER` do zwrócenia wyniku małymi literami. Wykorzystaj operator `(+)` do połączenia stringów.

```
SELECT LOWER(firstname+middleinitial+SUBSTRING(lastname, 1, 2)) AS 'lista
proponowanych loginów e-mail'
FROM member
WHERE lastname LIKE 'Anderson'
```

11. Napisz polecenie, które wybiera `title` i `title_no` z tablicy `title`.

- Wynikiem powinna być pojedyncza kolumna o formacie jak w przykładzie poniżej:

The title is: Poems, title number 7

- Czyli zapytanie powinno zwracać pojedynczą kolumnę w oparciu o wyrażenie, które łączy 4 elementy:
 - stała znakowa `.The title is:.`
 - wartość kolumny `title`
 - stała znakowa `.title number.`
 - wartość kolumny `title_no`

```
SELECT 'The title is: '+title+', '+cast(title_no AS char) AS titles
FROM title
```

Grupowanie, funkcje agregujące
Baza NORTHWIND

1. Policz średnią cenę jednostkową dla wszystkich produktów w tabeli products.

```
SELECT AVG(UnitPrice)
FROM Products
```

2. Zsumuj wszystkie wartości w kolumnie quantity w tabeli order details

```
SELECT SUM(quantity)
FROM [Order Details]
```

3. Podaj liczbę produktów o cenach mniejszych niż 10\$ lub większych niż 20\$

```
SELECT COUNT(*)
FROM Products
WHERE (UnitPrice < 10) OR (UnitPrice > 20)
```

4. Podaj maksymalną cenę produktu dla produktów o cenach poniżej 20\$

```
SELECT MAX(UnitPrice)
FROM Products
WHERE (UnitPrice < 20)
order by UnitPrice desc
```

5. Podaj maksymalną i minimalną i średnią cenę produktu dla produktów o produktach sprzedawanych w butelkach (bottle)

```
SELECT      MAX(UnitPrice) AS 'maks. cena produktów sprzedawanych w butelkach',
MIN(UnitPrice) AS 'min. cena produktów sprzedawanych w butelkach',
AVG(UnitPrice) AS 'śr. cena produktów sprzedawanych w butelkach'
FROM Products
WHERE QuantityPerUnit LIKE '%bottle%'
```

6. Wypisz informację o wszystkich produktach o cenie powyżej średniej

```
DECLARE @av decimal (12,2) --deklaracja zmiennej
SET @av = 45 --tu powinna być podstawiona średnia AVG(UnitPrice)
SELECT *
FROM Products
WHERE UnitPrice > @av
ORDER BY UnitPrice
```

7. Podaj sumę zamówienia o numerze 10250

```
select OrderId, sum(UnitPrice) AS Total_Price
from [Order Details]
where OrderId = 10250
group by OrderId
```

8. Napisz polecenie, które zwraca informacje o zamówieniach z tablicy order details. Zapytanie ma grupować i wyświetlać identyfikator każdego produktu a następnie obliczać ogólną zamówioną ilość. Ogólna ilość jest sumowana funkcją agregującą SUM i wyświetlana jako jedna wartość dla każdego produktu.

```
SELECT ProductID, SUM(Quantity) AS Total_Quantity
FROM [Order Details]
GROUP BY ProductID
```

9. Podaj maksymalną cenę zamawianego produktu dla każdego zamówienia

```
SELECT OrderID, MAX(UnitPrice) AS Maksymalna Cena
FROM [order details]
GROUP BY OrderID
```

10. Posortuj zamówienia wg maksymalnej ceny produktu

```
SELECT OrderID, MAX(UnitPrice) AS Max
FROM [Order Details]
GROUP BY OrderID
ORDER BY Max DESC
```

11. Podaj maksymalną i minimalną cenę zamawianego produktu dla każdego zamówienia

```
select OrderId, max(UnitPrice) as max, min(UnitPrice) as min
from [Order Details]
group by OrderId
```

12. Podaj liczbę zamówień dostarczanych przez poszczególnych spedytatorów

```
select ShipVia, count(OrderId) AS Total_Orders
from Orders
group by ShipVia
```

13. Który z spedytatorów był najaktywniejszy w 1997 roku

```
select ShipVia, count(OrderDate) AS 'Total Orders in 1997'
from Orders
where YEAR(OrderDate) = 1997
group by ShipVia
```

14. Wyświetl zamówienia dla których liczba pozycji zamówienia jest większa niż 5

```
select OrderID, count(OrderID) as liczba_pozycji_zamowienia
from [Order Details]
group by OrderID having count(OrderID) > 5
```

15. Wyświetl klientów którzy dla których w 1998 roku zrealizowano więcej niż 8 zamówień (wyniki posortuj malejąco wg łącznej kwoty za dostarczenie zamówień dla każdego z klientów)

```
select CustomerID, count(OrderID) AS 'ilosc zamownien w 1998', sum(Freight) AS
'łączna kwota za dostarczenie zamowien'
from Orders
where YEAR(OrderDate) = 1998
group by CustomerID having count(OrderID) > 8
order by sum(Freight) desc
```

Łączenie danych z wielu tabel

Baza NORTHWIND

1. Napisz polecenie zwracające nazwy produktów i firmy je dostarczające (tak aby produkty bez dostawców i dostawcy bez produktów nie pojawiali się w wyniku).

```
select ProductName, Suppliers.CompanyName
from Products
join Suppliers as Suppliers on Suppliers.SupplierID = Products.SupplierID
```

2. Napisz polecenie zwracające jako wynik nazwy klientów, którzy złożyli zamówienia po 01 marca 1998

```
SELECT DISTINCT CompanyName
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID AND
( (YEAR(Orders.OrderDate) = 1998 AND MONTH(Orders.OrderDate) > 2) OR YEAR(Orders.OrderDate)
> 1998 )
```

```
select distinct Customers.CompanyName
from Customers
join Orders on Orders.CustomerID = Customers.CustomerID
where Orders.OrderDate > '3/1/98'
```

3. Napisz polecenie zwracające wszystkich klientów z datami zamówień.

```
SELECT CompanyName, OrderDate
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
```

```
select Customers.CompanyName, Orders.OrderDate
from Customers
join Orders on Customers.CustomerID =Orders.CustomerID
```

Baza LIBRARY

1. 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 mj.firstname as FirstName, mj.lastname as LastName, birth_date, city, street, state,
ma.firstname as AdultFirstName, ma.lastname as AdultLastName
from juvenile
join member as mj on juvenile.member_no = mj.member_no
join adult on adult.member_no = juvenile.adult_member_no
join member as ma on ma.member_no = adult.member_no
```

2. Napisz polecenie, które podaje tytuły aktualnie wypożyczonych książek

```
select distinct title
from title
join loan on loan.title_no = title.title_no

select distinct title
from title
inner join copy on title.title_no = copy.title_no and on_loan = 'Y'
```

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 title, due_date, in_date, fine_assessed, fine_paid, datediff(d,in_date,due_date) as
przetrzymywana_dni
from loanhist join title on loanhist.title_no = title.title_no
where title like 'Tao Teh King' and in_date < due_date
```



```
order by fine_paid desc
```

DATEDIFF (datepart, startdate, enddate)

4. Napisz polecenie które podaje listę książek (numery ISBN) zarezerwowanych przez osobę o nazwisku: Stephen A. Graff

```
select reservation.isbn
from reservation
join member on member.member_no = reservation.member_no
where member.firstname like 'Stephen' and member.middleinitial like 'A' and member.lastname
like 'Graff'
```

Baza NORTHWIND

1. Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20.00 a 30.00, dla każdego produktu podaj dane adresowe dostawcy.

```
SELECT ProductName, UnitPrice, Address+' '+City+' '+PostalCode+' '+Country AS 'Adres'
FROM Products, Suppliers
WHERE Suppliers.SupplierID = Products.SupplierID AND (UnitPrice BETWEEN 20.00 AND 30.00)
```

2. Wybierz nazwy produktów oraz inf. o stanie magazynu dla produktów dostarczanych przez firmę Tokyo Traders.

```
SELECT ProductName, UnitsInStock
FROM Products, Suppliers
WHERE (Suppliers.SupplierID = Products.SupplierID) AND Suppliers.CompanyName LIKE 'Tokyo Traders'
```

3. Wybierz nazwy i numery telefonów dostawców, dostarczających produkty, których aktualnie nie ma w magazynie

```
SELECT CompanyName, Phone
FROM Products, Suppliers
WHERE (Suppliers.SupplierID = Products.SupplierID) AND Products.UnitsInStock = 0
```

4. Napisz polecenie zwracające listę produktów zamawianych w dniu 1996-07-08.

```
SELECT ProductName, OrderDate
FROM Products
JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
JOIN Orders ON Orders.OrderID = [Order Details].OrderID
WHERE YEAR(OrderDate) = 1996 AND MONTH(OrderDate) = 7 AND DAY(OrderDate) = 8
```

5. Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20.00 a 30.00, dla każdego produktu podaj dane adresowe dostawcy, interesują nas tylko produkty z kategorii Meat/Poultry.

```
SELECT ProductName, UnitPrice, Suppliers.Address, Categories.CategoryName
FROM Products
JOIN Suppliers ON Suppliers.SupplierID = Products.SupplierID
JOIN Categories ON Products.CategoryID = Categories.CategoryID
WHERE (UnitPrice BETWEEN 20 AND 30) AND Categories.CategoryName = 'Meat/Poultry'
```

6. Wybierz nazwy i ceny produktów z kategorii Confections dla każdego produktu podaj nazwę dostawcy.

```
SELECT ProductName, Categories.CategoryName, UnitPrice, Suppliers.CompanyName
FROM Products
JOIN Suppliers ON Suppliers.SupplierID = Products.SupplierID
JOIN Categories ON Products.CategoryID = Categories.CategoryID
WHERE Categories.CategoryName = 'Confections'
```

7. Wybierz nazwy i numery telefonów klientów, którym w 1997 roku przesyłki dostarczała firma United Package.

```
SELECT DISTINCT Customers.CompanyName, Customers.Phone
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN Shippers ON Shippers.ShipperID = Orders.ShipVia
WHERE YEAR(Orders.OrderDate) = 1997 AND Shippers.CompanyName LIKE 'United Package'
```

8. Wybierz nazwy i numery telefonów klientów, którzy kupowali produkty z kategorii Confections.

```
SELECT DISTINCT CompanyName, Phone
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
JOIN Products ON Products.ProductID = [Order Details].ProductID
JOIN Categories ON Categories.CategoryID = Products.CategoryID
WHERE CategoryName = 'Confections'
```

1. Napisz polecenie, które wyświetla listę wszystkich kupujących te same produkty. ???

```
select ProductName, CompanyName
from Customers
join Orders on Customers.CustomerID = Orders.CustomerID
join [Order Details] on [Order Details].OrderID = Orders.OrderID
join Products on Products.ProductID = [Order Details].ProductID
```

2. Napisz polecenie, które pokazuje pary pracowników zajmujących to samo stanowisko

```
SELECT a.LastName, a.Title, b.Title, b.LastName
FROM Employees AS a
JOIN Employees AS b
ON a.Title = b.Title
WHERE a.EmployeeID < b.EmployeeID
```

3. Napisz polecenie, które wyświetla pracowników oraz ich podwładnych

```
SELECT a.EmployeeID, a.FirstName, a.LastName, a.ReportsTo, b.EmployeeID, b.FirstName,
b.LastName, b.ReportsTo
FROM Employees AS a
JOIN Employees AS b ON b.ReportsTo = a.EmployeeID
ORDER BY a.LastName
```

4. Napisz polecenie, które wyświetla pracowników, którzy nie mają podwładnych

```
SELECT a.EmployeeID, a.FirstName, a.LastName, a.ReportsTo
FROM Employees AS a
LEFT OUTER JOIN Employees AS b ON b.ReportsTo = a.EmployeeID
WHERE b.EmployeeID is NULL
ORDER BY a.LastName
```

Baza LIBRARY

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

```
select distinct member.member_no, street, city, state, zip
from member
join adult on member.member_no = adult.member_no
join juvenile on juvenile.adult_member_no = member.member_no
where juvenile.birth_date < '1/1/96'
```

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 distinct member.member_no, street, city, state, zip
from member
join adult on member.member_no = adult.member_no
join juvenile on juvenile.adult_member_no = member.member_no
left join loan on member.member_no = loan.member_no
where juvenile.birth_date < '1/1/96' and isbn is null
```

3. 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 member
join adult on member.member_no = adult.member_no
```

4. Napisz polecenie, które zwraca: isbn, copy_no, on_loan, title, translation, cover, dla książek o isbn 1, 500 i 1000. Wynik posortuj wg ISBN

```
select item.isbn, title.title, copy_no, on_loan, translation, cover
from item
join title on title.title_no = item.title_no
join copy on copy.isbn = item.isbn
where item.isbn = 1 or item.isbn = 500 or item.isbn = 1000
order by item.isbn
```

5. Napisz polecenie które zwraca o użytkownikach biblioteki o nr 250, 342, i 1675 (dla każdego użytkownika: nr, imię i nazwisko członka biblioteki), oraz informację o zarezerwowanych książkach (isbn, data)

6. Podaj listę członków biblioteki mieszkających w Arizonie (AZ) mają więcej niż dwoje dzieci zapisanych do biblioteki

```
SELECT member.firstname, member.lastname, member.member_no, adult.state,
COUNT (*) AS Liczba_dzieci
FROM member
JOIN adult
ON adult.member_no = member.member_no
JOIN juvenile
ON member.member_no = juvenile.adult_member_no
WHERE adult.state = 'AZ'
GROUP BY member.firstname, member.lastname, member.member_no, adult.state
HAVING COUNT (*) > 2
```

Baza NORTHWIND

1. Dla każdej kategorii produktu, podaj łączną liczbę zamówionych jednostek

```
SELECT Categories.CategoryName, SUM(Products.UnitsOnOrder) AS Suma_Zamowionych_Jednostek
FROM Categories
JOIN Products ON Categories.CategoryID = Products.CategoryID
GROUP BY Categories.CategoryName
```

2. Dla każdego zamówienia podaj łączną liczbę zamówionych jednostek

```
SELECT [Order Details].OrderID, SUM(quantity) AS Sum_Quantity
FROM [Order Details]
JOIN Products ON [Order Details].ProductID = Products.ProductID
GROUP BY [Order Details].OrderID
ORDER BY 2 DESC
```

3. Zmodyfikuj poprzedni przykład, aby pokazać tylko takie zamówienia, dla których łączna liczba jednostek jest większa niż 250

```
SELECT [Order Details].OrderID, SUM(quantity) AS Sum_Quantity
FROM [Order Details]
```

```
JOIN Products ON [Order Details].ProductID = Products.ProductID
GROUP BY [Order Details].OrderID HAVING (SUM(quantity) > 250)
ORDER BY 2 DESC
```

Podzapytania

Baza NORTHWIND

1. Wybierz nazwy i numery telefonów klientów , którym w 1997 roku przesyłki dostarczała firma United Package.

```
select distinct cu.companyname, cu.phone from customers as cu
join orders on orders.customerid = cu.customerid
join shippers as sh on sh.shipperid = orders.shipvia
where year(orderdate) = 1997 and sh.companyname = 'United Package'
order by cu.companyname
```

```
select distinct CustomerID, CompanyName, Phone
from Customers
where CustomerID in (select CustomerID from Orders where YEAR(OrderDate) = 1997 and
ShipVia in (select ShipperID from Shippers where CompanyName = 'United Package'))
```

2. Wybierz nazwy i numery telefonów klientów, którzy kupowali produkty z kategorii Confections

```
select cu.CompanyName, cu.Phone
from Customers as cu
where cu.CustomerID in (select c.CustomerID
from Customers as c
join orders as o on o.CustomerID = c.CustomerID
join [Order Details] as od on o.OrderID = od.OrderID
join Products as p on p.ProductID = od.ProductID
join Categories as cat on cat.CategoryID = p.CategoryID
where cat.CategoryName like 'Confections')
order by cu.CompanyName
```

3. Wybierz nazwy i numery telefonów klientów, którzy nie kupili żadnego produktu z kategorii .Confections.

```
select cu.CompanyName, cu.Phone
from Customers as cu
where cu.CustomerID not in (select c.CustomerID
from Customers as c
join orders as o on o.CustomerID = c.CustomerID
join [Order Details] as od on o.OrderID = od.OrderID
join Products as p on p.ProductID = od.ProductID
join Categories as cat on cat.CategoryID = p.CategoryID
where cat.CategoryName like 'Confections')
order by cu.CompanyName
```

4. Wybierz nazwy i numery telefonów klientów, którzy kupili więcej niż 3 różne produkty z kategorii .Confections.

5. Dla każdego produktu podaj maksymalną liczbę zamówionych jednostek

6. Podaj wszystkie produkty których cena jest mniejsza niż średnia cena produktu

7. Podaj wszystkie produkty których cena jest mniejsza niż średnia cena produktu danej kategorii

8. Dla każdego produktu podaj jego nazwę, cenę, średnią cenę wszystkich produktów oraz różnicę między ceną produktu a średnią ceną wszystkich produktów

9. Dla każdego produktu podaj jego nazwę kategorii, nazwę produktu, cenę, średnią cenę wszystkich produktów danej kategorii oraz różnicę między ceną produktu a średnią ceną wszystkich produktów danej kategorii.

10. Podaj łączną wartość zamówienia o numerze 1025 (uwzględnij cenę za przesyłkę).
11. Podaj łączną wartość zamówień każdego zamówienia (uwzględnij cenę za przesyłkę).
12. Czy są jacyś klienci którzy nie złożyli żadnego zamówienia w 1997 roku, jeśli tak to pokaż ich dane adresowe.

```
select cu.CompanyName, cu.Address, cu.City
from Customers as cu
where cu.CustomerID not in (select c.CustomerID
                           from Customers as c
                           left join Orders as o on c.CustomerID = o.CustomerID
                           where YEAR(o.OrderDate) = 1997)
```

13. Podaj produkty kupowane przez więcej niż jednego klienta
14. Podaj produkty kupowane przez więcej niż 20 klientów

```
select T.ProductName
from (select ProductName, COUNT(CustomerID) AS Ile
      from [Order Details] as od
      join Products as pr on pr.ProductID = od.ProductID
      join orders as o on o.orderid = od.orderid
      group by ProductName having COUNT(CustomerID) > 20) AS T
```

1. Napisz polecenie wyświetlające imię i nazwisko czytelnika oraz miejsce jego zamieszkania. Dla każdego podaj ile ma aktualnie pożyczonych książek.

- jeżeli nie ma aktualnie żadnej pożyczonej książki – w wyniku powinno wyświetlić się 0.
- zmodyfikuj zapytanie tak, żeby było wiadomo czy czytelnik jest osobą dorosłą, czy dzieckiem

//pewnie da się krocej lub sprytniej

```
select a.member_no, m.firstname, m.lastname, a.street, a.city, 'Adult' as S, count(copy_no)
as [ile ksiazek]
from adult as a
join member as m on m.member_no = a.member_no
left join loan as l on l.member_no = m.member_no
group by a.member_no, m.firstname, m.lastname, a.street, a.city
union
select j.member_no, me.firstname, me.lastname, ad.street, ad.city, 'Child' as S,
count(copy_no) as [ile ksiazek]
from juvenile as j join adult as ad on ad.member_no = j.adult_member_no
join member as me on me.member_no = j.member_no
left join loan as l on l.member_no = me.member_no
group by j.member_no, me.firstname, me.lastname, ad.street, ad.city

order by [ile ksiazek] desc
```

2. Napisz polecenie wyświetlające imię i nazwisko czytelnika oraz miejsce jego zamieszkania. Dla każdego z nich podaj ile pożyczył książek w ostatnim (trzecim... itd.) kwartale 2005 roku. Jeżeli nie pożyczył żadnej książki – powinno pojawić się 0.

3. Czy są jacyś klienci, którzy nie złożyli żadnego zamówienia w 1996 roku. Rozwiąż zadanie dwoma sposobami: przy użyciu mechanizmów zapytań i bez użycia.

```
SELECT DISTINCT Customers.CompanyName
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID = Orders.CustomerID AND YEAR(OrderDate) = 1996
WHERE OrderID is NULL
```

```
SELECT DISTINCT Customers.CompanyName
FROM Customers
LEFT JOIN (SELECT DISTINCT CustomerID, OrderID FROM Orders WHERE YEAR(OrderDate) = 1996) AS
o ON o.CustomerID = Customers.CustomerID
WHERE OrderID is NULL
```

4. Dla każdego pracownika podaj jego imię i nazwisko oraz:

- całkowitą liczbę zamówień, które obsłużyli
- z rozbiciem na lata, kwartały i miesiące

5. Który z pracowników w 1997 r. obsłużył zamówienia o najwyższej wartości (wliczając rabat oraz opłatę za przesyłkę). Podaj jego imię i nazwisko.

```
select X.FirstName+' '+X.LastName as [Pracownik roku 1997]
from
(select top 1 T.FirstName, T.LastName, T.OrderID, T.Freight, SUM(T.Wartosc) as [Wartosc
zamowienia]
from (select e.FirstName, e.LastName, od.OrderID, o.Freight, (od.UnitPrice*od.Quantity)-
(od.UnitPrice*od.Quantity*od.Discount) as Wartosc
from Employees as eS
join Orders as o on e.EmployeeID = o.EmployeeID
join [Order Details] as od on o.OrderID = od.OrderID
where YEAR(o.OrderDate) = 1997) AS T
group by T.FirstName, T.LastName, T.OrderID, T.Freight
order by SUM(T.Wartosc)-T.Freight desc) AS X
```

6. Czy są jacyś klienci, którzy w 1996 roku nie kupili produktu z kategorii 'Confections'?

7. Wyświetl pracowników, którzy nie mają podwładnych i liczbę obsłużonych przez każdego z nich zamówień w 1997.

```
select T.EmployeeID, T.LastName, COUNT(o.OrderDate) AS [liczba obsluzonych zamowien w 1997]
from (select e1.EmployeeID, e1.LastName
      from Employees as e1
      left outer join Employees as e2 on e2.ReportsTo = e1.EmployeeID
      where e2.ReportsTo is null) as T
join Orders as o on o.EmployeeID = T.EmployeeID
where YEAR(o.OrderDate) = 1997
group by T.EmployeeID, T.LastName
order by T.EmployeeID
```

8. Który klient zamówił produkty o największej łącznej wartości w 1996 roku - z rozbiem na lata (-- polecenie rozbięcia na lata jest akurat mało potrzebne :) --) i miesiące

9. Który czytelnik zapłacił największą karę w którymś tam roku. (tu uwaga, że Marcjanowi nie spodobało się podanie samego member_no, czyli trzeba łączyć z tabelą member).

10. Podaj dla każdego produktu z kategorii 'Confections' liczbę produktów zamówionych w marcu 1997.

```
select T.ProductName, SUM(od.Quantity) AS [Quantity 3/1997]
from (select p.ProductID, p.ProductName
      from Products as p
      join Categories as c on p.CategoryID = c.CategoryID
      where c.CategoryName like 'Confections') AS T
join [Order Details] as od on T.ProductID = od.ProductID
join Orders as o on o.OrderID = od.OrderID
where YEAR(o.OrderDate) = 1997 AND MONTH(o.OrderDate) = 3
group by T.ProductName
order by T.ProductName
```

```
select p.ProductName, SUM(od.Quantity) AS [Quantity 3/1997]
from Products as p
join Categories as c on p.CategoryID = c.CategoryID
join [Order Details] as od on od.ProductID = p.ProductID
join Orders as o on o.OrderID = od.OrderID
where c.CategoryName like 'Confections' AND YEAR(o.OrderDate) = 1997 AND MONTH(o.OrderDate) = 3
group by p.ProductName
order by p.ProductName
```

11. Zapytanie zwraca listę wszystkich pracowników którzy obsłużyli zamówienie w '9/5/97'.

```
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS (SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid AND o.orderdate = '9/5/97')
```

```
SELECT lastname, employeeid
FROM employees AS e
WHERE employeeid in (SELECT employeeid FROM orders AS o WHERE o.orderdate = '9/5/97')
```

```
SELECT DISTINCT lastname, e.employeeid
FROM orders AS o
INNER JOIN employees AS e ON o.employeeid = e.employeeid
WHERE o.orderdate = '9/5/1997'
```