

-
1. Napisz instrukcję SELECT, która zwróci kolumny custid, companyname, contactname, address, city, country, phone z tabeli Sales.Customers. Filtruj wyniki, aby uwzględnić tylko klientów z kraju Brazylia.

-
2. Napisz instrukcję SELECT, która zwróci kolumny custid, companyname, contactname, address, city, country, phone z tabeli Sales.Customers. Filtruj wyniki, aby uwzględnić tylko klientów z Brazyli, UK, USA.

-
3. Napisz instrukcję SELECT, która zwróci kolumny custid, companyname, contactname, address, city, country, phone z tabeli Sales.Customers. Filtruj wyniki, aby uwzględnić tylko klientów kolumny contactname rozpoczynającej się od litery A.

-
4. Dział IT napisał instrukcję T-SQL, która pobiera kolumny custid i nazwa firmy z tabeli Sales.Customers i kolumny orderid z tabeli Sales.Orders.

Wykonaj zapytanie. Zwróć uwagę na dwie rzeczy. Po pierwsze, zapytanie pobiera wszystkie wiersze z tabeli Sales.Customers. Po drugie, istnieje operator porównania w klauzuli ON, określający, że kolumna city powinna być równa wartości "Paris".

Skopiuj podaną instrukcję T-SQL i zmodyfikuj ją tak, aby mieć operator porównania dla kolumny city w klauzuli WHERE. Wykonaj zapytanie.

Czy wynik jest taki sam jak w pierwszym teście T-SQL? Czemu? Jaka jest różnica między określaniem predykatu w klauzuli ON i w klauzuli WHERE?

SELECT

c.custid, c.companyname, o.orderid

FROM Sales.Customers AS c

LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid AND c.city= 'Paris';

5. Napisz instrukcję T-SQL, aby pobrać klientów (customers) z tabeli Sales.Customers, którzy nie mają pasujących zamówień w tabeli Sales.Orders. Dopasowanie klientów do zamówień opiera się na porównaniu między custid (dla Sales.Customers) a wartością zamówienia (custid z Sales.Orders). Odzyskaj kolumny custid i nazwa firmy z tabeli Sales.Customers. (Wskazówka: użyj instrukcji T-SQL, która jest podobna do instrukcji z poprzedniego zadania.)

6. Napisz instrukcję SELECT, aby pobrać kolumny custid i contactname z tabeli Sales.Customers oraz kolumny orderid i orderdate z tabeli Sales.Orders. Filtruj wyniki, aby uwzględnić tylko zamówienia złożone w/po 1 kwietnia 2008 r. (Filtruj kolumnę orderdate). Następnie posortuj wynik według orderdate w porządku malejącym i custid w porządku rosnącym.
-

7. Wykonaj zapytanie dokładnie tak, jak zostało napisane w oknie zapytania i obserwuj wynik.

Otrzymujesz błąd. Jaki jest komunikat o błędzie? Jak myślisz, dlaczego masz ten błąd? (Wskazówka: Zapamiętaj logiczny porządek przetwarzania zapytania.)

Zastosuj potrzebne zmiany do instrukcji SELECT, aby działały bezbłędnie. Przetestuj zmiany, wykonując instrukcję T-SQL.

SELECT

e.empid, e.lastname, e.firstname, e.title, e.mgrid,

m.lastname AS mgrlastname, m.firstname AS mgrfirstname

FROM HR.Employees AS e

INNER JOIN HR.Employees AS m ON e.mgrid = m.empid

WHERE

mgrlastname = N'Buck';

8. Skopiuj istniejącą instrukcję T-SQL z poprzedniego zadania i zmodyfikuj ją tak, aby wynik zwrócił wszystkich pracowników i został posortowany przez imię menedżera. Spróbuj najpierw użyć nazwy kolumny źródłowej, a następnie spróbuj użyć nazwy kolumny aliasu.

Dlaczego można użyć nazwy kolumny źródłowej lub nazwy kolumny aliasu?

9. Napisz instrukcję SELECT do tabeli Sales.Orders i pobierz kolumny orderid i orderdate. Wyszukaj 20 ostatnich zamówień w oparciu o kolejność zamówień.
-

10. Napisz instrukcję SELECT, aby pobrać ten sam wynik co w poprzednim zadaniu, ale użyj klauzuli OFFSET-FETCH.

Wykonaj pisemne oświadczenie i porównaj uzyskane wyniki z wynikami poprzedniego zadania.

11. Napisz instrukcję SELECT, aby pobrać kolumny productname i unitprice z tabeli Production.Products.

Wykonaj instrukcję T-SQL i zwróć uwagę na liczbę zwróconych wierszy.

Zmodyfikuj instrukcję SELECT tak, aby obejmowała tylko 10% produktów na podstawie kolejności kolumny unitprice.

Czy jest możliwe wykonanie tego zadania z klauzulą OFFSET-FETCH?

12. Napisz instrukcję SELECT, aby pobrać kolumny custid, orderid i orderdate z tabeli Sales.Orders. Posortuj wiersze według daty zamówienia (orderdate) i zamówienia (orderid). Wyszukaj pierwsze 20 wierszy.
-

13. Skopiuj instrukcję SELECT do zadania poprzedniego i zmodyfikuj klauzulę OFFSET-FETCH, aby pominąć pierwsze 20 wierszy i pobrać następne 20 wierszy.
-

14. Napisz instrukcję SELECT, aby zwrócić kolumny zawierające:

- Aktualna data i czas. Użyj aliasu currentdatetime.
- Tylko aktualna data. Użyj aliasu currentdate.
- Tylko aktualny czas. Użyj aliasu currenttime.
- Tylko w bieżącym roku. Użyj aliasu currentyear.
- Tylko numer bieżącego miesiąca. Użyj aliasu currentmonth.
- Tylko numer bieżącego dnia miesiąca. Użyj aliasu currentday.
- Tylko numer bieżącego tygodnia w roku. Użyj aliasu currentweeknumber.
- Nazwa bieżącego miesiąca na podstawie kolumny currentdatetime. Użyj aliasu currentmonthname.

Czy możesz użyć aliasu currentdatetime jako źródła w drugiej kolumnie (currentdate)? Proszę wytłumaczyć.

15. Napisz 11 grudnia 2011 r. jako kolumnę z typem daty. Skorzystaj z różnych możliwości języka T-SQL (cast, convert, specific function, etc.) I użyj aliasu "somedate".

16. Napisz instrukcję SELECT, aby zwrócić kolumny zawierające:

- Trzy miesiące [threemonths] od bieżącej daty i godziny. Użyj aliasów.
 - Liczba dni między bieżącą datą a pierwszą kolumną [threemonths]. Użyj aliasu diffdays.
 - Liczba tygodni między 4 kwietnia 1992 a 16 września 2011. Użyj aliasów.
 - Pierwszy dzień w bieżącym miesiącu na podstawie bieżącej daty i godziny. Najpierw użyj aliasu.
-

17. Dział IT napisał instrukcję T-SQL, która tworzy i wypełnia tabelę o nazwie Sales.Somedates.

Wykonaj dostarczoną instrukcję T-SQL.

Napisz instrukcję SELECT do tabeli Sales.Somedates i pobierz kolumnę isitdate. Dodaj nową kolumnę o nazwie convertdate z nową wartością typu danych daty na podstawie kolumny isitdate. Jeśli nie można przekształcić kolumny isitdate w typ danych daty dla określonego wiersza, zwróć wartość NULL.

Jaka jest różnica między funkcjami SYSDATETIME i CURRENT_TIMESTAMP?

Co to jest format neutralny językowo dla typu DATE?

```
SET NOCOUNT ON;
```

```
IF OBJECT_ID('Sales.Somedates') IS NOT NULL
```

```
    DROP TABLE Sales.Somedates;
```

```
CREATE TABLE Sales.Somedates (
```

```
    isitdate varchar(9)
```

```
);
```

```
INSERT INTO Sales.Somedates (isitdate) VALUES
```

```
    ('20110101'),
```

```
    ('20110102'),
```

```
    ('20110103X'),
```

```
('20110104'),  
('20110105'),  
('20110106'),  
('20110107Y'),  
('20110108');
```

```
SET NOCOUNT OFF;
```

```
SELECT isitdate
```

```
FROM Sales.Somedates;
```

18. Wykonaj:

```
DROP TABLE Sales.Somedates;
```

19. Napisz instrukcję SELECT, aby pobrać różne wartości dla kolumny custid z tabeli Sales.Orders. Filtruj wyniki, aby uwzględnić tylko zamówienia złożone w lutym 2008 r.

20. Napisz instrukcję SELECT z tymi kolumnami:

- Aktualna data i czas
- Pierwsza data bieżącego miesiąca
- Ostatnia data bieżącego miesiąca

21. Napisz instrukcję SELECT do tabeli Sales.Orders i pobierz kolumny orderid, custid i orderdate. Filtruj wyniki, aby uwzględnić tylko zamówienia złożone w ciągu ostatnich pięciu dni miesiąca zamówienia.

22. Napisz instrukcję SELECT na tabelach Sales.Orders i Sales.OrderDetails i pobierz wszystkie unikatowe wartości (użyj DISTINCT) dla kolumny productid. Filtruj wyniki, aby uwzględnić tylko zamówienia złożone w ciągu pierwszych 10 tygodni roku 2007.

23. Napisz instrukcję SELECT do tabeli Sales.Customers i pobierz kolumny contactname i city. Połącz oba kolumny, aby nowa kolumna wyglądała następująco:

Allen, Michael (miasto: Berlin)

24. Skopiuj instrukcję T-SQL do poprzedniego zadania i zmodyfikuj ją, aby rozszerzyć kolumnę o nowe informacje z kolumny region. Traktuj wartość NULL w kolumnie region jako pusty ciąg znaków w celu połączenia. Kiedy region ma wartość NULL, zmodyfikowana kolumna powinna wyglądać następująco:

Allen, Michael (miasto: Berlin, region:)

Gdy region nie ma wartości NULL, zmodyfikowana kolumna powinna wyglądać następująco

Richardson, Shawn (miasto: Sao Paulo, region: SP)

25. Napisz instrukcję SELECT, aby pobrać kolumny contactname i contacttitle z tabeli Sales.Customers. Zwróć tylko wiersze, w których pierwszy znak w contactname to "A" do "G".
-

26. Napisz instrukcję SELECT, aby pobrać kolumnę contactname z tabeli Sales.Customers. Na podstawie tej kolumny dodaj kolumnę o nazwie lastname, która powinna składać się ze wszystkich znaków przed przecinkiem.
-

27. Napisz SELECT, aby pobrać kolumnę ContactName z tabeli Sales.Customers i zastąp przecinek w contactname pustym ciągiem. Na podstawie tej kolumny dodaj kolumnę o nazwie firstname, która powinna składać się ze wszystkich znaków po przecinku.
-

28. Napisz instrukcję SELECT, aby pobrać custid z tabeli Sales.Customers. Na podstawie kolumny custid stwórz: custid o stałej szerokości (6 znaków) poprzedzonego literą C i zerami. Na przykład wartość custid 1 powinna wyglądać tak C00001 a custid 10 C00010.
-

29. Napisz instrukcję SELECT, aby pobrać kolumnę contactname z tabeli Sales.Customers. Dodaj kolumnę, która powinna policzyć liczbę pojawienia się znaku "a" w contactname. (Wskazówka: Użyj funkcji REPLACE i LEN.) Posortuj wynik od najwyższej do najniższej liczby pojawiania się znaku "a".
-

30. Napisz instrukcję SELECT do tabeli Production.Products, aby stworzyć kolumnę o nazwie productdesc. Kolumna powinna opierać się na kolumnach productname i unitprice i wyglądać tak:

Cena jednostkowa produktu HHYDP to 18,00 \$.

Czy używałeś funkcji CAST lub CONVERT? Który z nich jest bardziej odpowiedni do użycia?

31. Departament marketingu USA dostarczył Ci datę rozpoczęcia 4/1/2007 (przy użyciu formatu US, czytaj jako 1 kwietnia 2007 r.) i datę końcową 11/30/2007 (używając formatu US, 30 listopada 2007 r.)). Napisz instrukcję SELECT do tabeli Sales.Orders, aby pobrać kolumny orderid, orderdate, shippeddate i shipregion. Przefiltruj wynik tak, aby zawierał tylko wiersze z datą zamówienia między określoną datą rozpoczęcia i datą zakończenia i miał więcej niż 30 dni między datą wysłania a datą zamówienia. Sprawdź również kolumnę shipregion pod kątem brakujących wartości. Jeśli brakuje wartości, zwróć wartość "No region".

W tej instrukcji SELECT można użyć funkcji CONVERT lub funkcję PARSE.

32. Dział IT chciałby zamienić wszystkie informacje o numerach telefonów w tabeli Sales.Customers na wartości całkowite. Dział IT wskazał, że wszystkie łączniki, nawiasy i spacje muszą zostać usunięte przed konwersją na typ danych całkowity (integer).

Napisz instrukcję SELECT, aby zrealizować wymagania działu IT. Zastąp wszystkie określone znaki w kolumnie (phone) tabeli Sales.Customers, a następnie przekonwertuj kolumnę z typu danych nvarchar na typ danych int. Instrukcja T-SQL nie może zawieść, jeśli wystąpił błąd konwersji, ale powinna zwrócić wartość NULL. (Podpowiedź: Najpierw spróbuj napisać instrukcję T-SQL za pomocą funkcji CONVERT, a następnie użyj nowszej funkcji poznanej przez Ciebie). Użyj aliasów dla tej kolumny PhoneAsIntiger.

33. Napisz instrukcję SELECT do tabeli Sales.Customers i pobierz kolumny custid i contactname. Dodaj kolumnę o nazwie segmentgroup za pomocą funkcji logicznej IIF o wartości "Target group" dla klientów z Meksyku i którzy w tytule (contacttitle) mają słowo "Owner". Użyj wartości "Other" dla reszty klientów.
-

34. Zmodyfikuj instrukcję T-SQL z poprzedniego zadania, aby wyświetlić w "Target group" wszystkich klientów, dla których region ma jakąś wartość lub słowo "Owner" w contacttitle.
-

35. Napisz instrukcję SELECT do tabeli Sales.Customers i pobierz kolumny custid i contactname. Dodaj kolumnę o nazwie segmentgroup, która za pomocą funkcji logicznej CHOOSE z

czterema możliwymi opisami ("Group One", "Group Two", "Group Three", "Group Four"). Użyj operatora modulo w kolumnie custid. (Użyj wyrażenia $\text{custid} \% 4 + 1$, aby określić grupę docelową.)

36. Napisz instrukcję SELECT, aby pobrać kolumny contactname i fax z tabeli Sales.Customers. Jeśli w kolumnie fax jest brak wartości, należy zwrócić "No Information".

Napisz dwa rozwiązania, jeden za pomocą funkcji COALESCE, a drugi za pomocą funkcji ISNULL.

Jaka jest różnica między funkcjami ISNULL i COALESCE?

37. Zaktualizuj podaną instrukcję T-SQL klauzulą WHERE, aby przefiltrować kolumnę region przy użyciu podanej zmiennej @region, która może mieć wartość lub wartość NULL. Przetestuj rozwiązanie.

```
DECLARE @region AS NVARCHAR(30) = NULL;
```

```
SELECT
```

```
    custid, region
```

```
FROM Sales.Customers;
```

```
GO
```

```
DECLARE @region AS NVARCHAR(30) = N'WA';
```

```
SELECT
```

```
    custid, region
```

```
FROM Sales.Customers;
```

38. Napisz instrukcję SELECT, aby pobrać kolumny contactname, city, region z tabeli Sales.Customers. Zwróć tylko wiersze, które nie mają dwóch znaków w kolumnie regionu lub gdzie region ma wartość NULL.
