

Bazy Danych – kartkówka zaliczeniowa – 19.12.2011 Imię, nazwisko:

Z podanych zadań wybierz i rozwiąż maksymalnie 5 zadań. Każde zadanie warte jest 10 punktów.

Punktacja zadania:

- 0 punktów – brak rozwiązania lub rozwiązanie całkowicie niepoprawne i znaczne błędy składniowe świadczące o nieznajomości języka SQL
- 3 punkty – rozwiązanie częściowo poprawne i liczne błędy składniowe
- 6 punktów – rozwiązanie częściowo poprawne i brak błędów składniowych lub rozwiązanie poprawne ale liczne błędy składniowe
- 9 punktów – rozwiązanie poprawne i nieliczne błędy składniowe
- 10 punktów – rozwiązane poprawne, składnia poprawna.

1. Napisz zapytanie, które wybierze z bazy danych Northwind trzy zamówienia o największym łącznym koszcie zamówionych towarów i wypisze kto złożył te zamówienia (CompanyName z tabeli Customers), kiedy zostało złożone, ile zamówiono towarów (suma pól Quantity) i na jaką łączną kwotę (Quantity*UnitPrice).

Hint: wykorzystaj tabele: Customers, Orders oraz [Order Details]

2. Napisz zapytanie, które z tabeli produktów wybierze te, których cena jest niższa niż średnia cen wszystkich produktów i przepisze je do nowej tabeli o nazwie CheapProducts.

Hint: wykorzystaj tabele: Products

3. Stwórz tabelę Struktura zawierającą pola: id, nazwisko (varchar), stopien (varchar), idPrzelozonego. Id jest kluczem głównym, nazwisko i stopien nie mogą być puste, stopień musi być jednym z napisów z listy („PVT”, „PV2”, „PFC”, „SPC”, „CPL”, „SGT”, „SSG”, „SFC”, „MSG”, „ISG”, „SGM”, „CSM”, „SMA”), idPrzelozonego może być NULL, jest kluczem obcym do tej samej tabeli (Struktura).

Wstaw do tabeli następujące dane:

```
(1, Smith, „SGM”, NULL)
(2, Celko, „MSG”, NULL)
(3, Ramirez, „SFC”, 2)
(4, Kovalsky, „SSG”, 1)
(5, Plateu, „SGT”, 1)
(6, Gore, „CPL”, 5)
```

Dla każdej osoby z tabeli Employees z bazy danych Northwind wstaw jej nazwisko (LastName) jako nazwisko, stopien jako „PVT” oraz przelozonego o id 3 dla osób których nazwisko kończy się na „n”, a 4 dla pozostałych osób.

4. Dla tabeli Struktura jak w zadaniu 3 napisz zapytanie (używając CTE), które zwróci wszystkie osoby które są podwładnymi (bezpośrednimi i pośrednimi) osoby o nazwisku „Smith”.
5. Zaproponuj rozwiązanie pamiętające zmiany w stopniach dla osób z tabeli Schemat z zadania 3. Niech będzie to nowa tabela, która będzie pamiętać, kto awansował (idOsoby) na jaki stopień (stopien), kiedy (dataOtrzymania). Usuń kolumnę stopien z tabeli Schemat. Stwórz widok, który zwraca wszystkie osoby, wraz z aktualnie posiadanym stopniem.
6. Załóżmy że, mamy daną tabelę pamiętającą awans zawodowy osób jak w zadaniu 5. Stwórz funkcję, która dla podanej jako parametr osoby (poprzez id osoby) zwraca jej aktualny stopień (jeśli wykonałeś zadanie 5 możesz założyć, że stworzony tam widok jest dostępny).

Tabele bazy danych Northwind:

```
CREATE TABLE [Customers](
    [CustomerID] [nchar](5) NOT NULL PRIMARY KEY,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
)
CREATE TABLE [Employees](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [LastName] [nvarchar](20) NOT NULL,
    [FirstName] [nvarchar](10) NOT NULL,
    [Title] [nvarchar](30) NULL,
    [TitleOfCourtesy] [nvarchar](25) NULL,
    [BirthDate] [datetime] NULL,
    [HireDate] [datetime] NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [HomePhone] [nvarchar](24) NULL,
    [Extension] [nvarchar](4) NULL,
    [Photo] [image] NULL,
    [Notes] [ntext] NULL,
    [ReportsTo] [int] NULL,
    [PhotoPath] [nvarchar](255) NULL,
)
CREATE TABLE [Orders](
    [OrderID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [CustomerID] [nchar](5) NULL,
    [EmployeeID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [RequiredDate] [datetime] NULL,
    [ShippedDate] [datetime] NULL,
    [ShipVia] [int] NULL,
    [Freight] [money] NULL,
    [ShipName] [nvarchar](40) NULL,
    [ShipAddress] [nvarchar](60) NULL,
    [ShipCity] [nvarchar](15) NULL,
    [ShipRegion] [nvarchar](15) NULL,
    [ShipPostalCode] [nvarchar](10) NULL,
    [ShipCountry] [nvarchar](15) NULL,
)
CREATE TABLE [Products](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](40) NOT NULL,
    [SupplierID] [int] NULL,
    [CategoryID] [int] NULL,
    [QuantityPerUnit] [nvarchar](20) NULL,
    [UnitPrice] [money] NULL,
    [UnitsInStock] [smallint] NULL,
    [UnitsOnOrder] [smallint] NULL,
    [ReorderLevel] [smallint] NULL,
    [Discontinued] [bit] NOT NULL,
)
CREATE TABLE [Order Details](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [Quantity] [smallint] NOT NULL,
    [Discount] [real] NOT NULL,
)
ALTER TABLE [Employees] ADD CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Customers] FOREIGN KEY([CustomerID])
REFERENCES [Customers] ([CustomerID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Employees] FOREIGN KEY([EmployeeID])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Orders] FOREIGN KEY([OrderID])
REFERENCES [Orders] ([OrderID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Products] FOREIGN KEY([ProductID])
REFERENCES [Products] ([ProductID])
```

Bazy Danych – kartkówka zaliczeniowa – 19.12.2011 Imię, nazwisko:

Z podanych zadań wybierz i rozwiąż maksymalnie 5 zadań. Każde zadanie warte jest 10 punktów.

Punktacja zadania:

- 0 punktów – brak rozwiązania lub rozwiązanie całkowicie niepoprawne i znaczne błędy składniowe świadczące o nieznajomości języka SQL
- 3 punkty – rozwiązanie częściowo poprawne i liczne błędy składniowe
- 6 punktów – rozwiązanie częściowo poprawne i brak błędów składniowych lub rozwiązanie poprawne ale liczne błędy składniowe
- 9 punktów – rozwiązanie poprawne i nieliczne błędy składniowe
- 10 punktów – rozwiązanie poprawne, składnia poprawna.

1. Napisz zapytanie, które wybierze z bazy danych Northwind trzy zamówienia o największym łącznym koszcie zamówionych towarów i wypisze kto złożył te zamówienia (CompanyName z tabeli Customers), kiedy zostało złożone, ile zamówiono towarów (suma pól Quantity) i na jaką łączną kwotę (Quantity*UnitPrice).

Hint: wykorzystaj tabele: Customers, Orders oraz [Order Details]

2. Napisz zapytanie, które z tabeli produktów wybierze te, których cena jest niższa niż średnia cen wszystkich produktów i przepisze je do nowej tabeli o nazwie CheapProducts.

Hint: wykorzystaj tabele: Products

3. Stwórz tabelę Struktura zawierającą pola: id, nazwisko (varchar), stopien (varchar), idPrzelozonego. Id jest kluczem głównym, nazwisko i stopien nie mogą być puste, stopień musi być jednym z napisów z listy („PVT”, „PV2”, „PFC”, „SPC”, „CPL”, „SGT”, „SSG”, „SFC”, „MSG”, „ISG”, „SGM”, „CSM”, „SMA”), idPrzelozonego może być NULL, jest kluczem obcym do tej samej tabeli (Struktura).

Wstaw do tabeli następujące dane:

```
(1, Smith, „SGM”, NULL)
(2, Celko, „MSG”, NULL)
(3, Ramirez, „SFC”, 2)
(4, Kovalsky, „SSG”, 1)
(5, Plateu, „SGT”, 1)
(6, Gore, „CPL”, 5)
```

Dla każdej osoby z tabeli Employees z bazy danych Northwind wstaw jej nazwisko (LastName) jako nazwisko, stopien jako „PVT” oraz przelozonego o id 3 dla osób których nazwisko kończy się na „n”, a 4 dla pozostałych osób.

4. Dla tabeli Struktura jak w zadaniu 3 napisz zapytanie (używając CTE), które zwróci wszystkie osoby które są podwładnymi (bezpośrednimi i pośrednimi) osoby o nazwisku „Smith”.
5. Zaproponuj rozwiązanie pamiętające zmiany w stopniach dla osób z tabeli Schemat z zadania 3. Niech będzie to nowa tabela, która będzie pamiętać, kto awansował (idOsoby) na jaki stopień (stopien), kiedy (dataOtrzymania). Usuń kolumnę stopien z tabeli Schemat. Stwórz widok, który zwraca wszystkie osoby, wraz z aktualnie posiadanym stopniem.
6. Załóżmy że, mamy daną tabelę pamiętającą awans zawodowy osób jak w zadaniu 5. Stwórz funkcję, która dla podanej jako parametr osoby (poprzez id osoby) zwraca jej aktualny stopień (jeśli wykonałeś zadanie 5 możesz założyć, że stworzony tam widok jest dostępny).

Tabele bazy danych Northwind:

```
CREATE TABLE [Customers](
    [CustomerID] [nchar](5) NOT NULL PRIMARY KEY,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
)
CREATE TABLE [Employees](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [LastName] [nvarchar](20) NOT NULL,
    [FirstName] [nvarchar](10) NOT NULL,
    [Title] [nvarchar](30) NULL,
    [TitleOfCourtesy] [nvarchar](25) NULL,
    [BirthDate] [datetime] NULL,
    [HireDate] [datetime] NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [HomePhone] [nvarchar](24) NULL,
    [Extension] [nvarchar](4) NULL,
    [Photo] [image] NULL,
    [Notes] [ntext] NULL,
    [ReportsTo] [int] NULL,
    [PhotoPath] [nvarchar](255) NULL,
)
CREATE TABLE [Orders](
    [OrderID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [CustomerID] [nchar](5) NULL,
    [EmployeeID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [RequiredDate] [datetime] NULL,
    [ShippedDate] [datetime] NULL,
    [ShipVia] [int] NULL,
    [Freight] [money] NULL,
    [ShipName] [nvarchar](40) NULL,
    [ShipAddress] [nvarchar](60) NULL,
    [ShipCity] [nvarchar](15) NULL,
    [ShipRegion] [nvarchar](15) NULL,
    [ShipPostalCode] [nvarchar](10) NULL,
    [ShipCountry] [nvarchar](15) NULL,
)
CREATE TABLE [Products](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](40) NOT NULL,
    [SupplierID] [int] NULL,
    [CategoryID] [int] NULL,
    [QuantityPerUnit] [nvarchar](20) NULL,
    [UnitPrice] [money] NULL,
    [UnitsInStock] [smallint] NULL,
    [UnitsOnOrder] [smallint] NULL,
    [ReorderLevel] [smallint] NULL,
    [Discontinued] [bit] NOT NULL,
)
CREATE TABLE [Order Details](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [Quantity] [smallint] NOT NULL,
    [Discount] [real] NOT NULL,
)
ALTER TABLE [Employees] ADD CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Customers] FOREIGN KEY([CustomerID])
REFERENCES [Customers] ([CustomerID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Employees] FOREIGN KEY([EmployeeID])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Orders] FOREIGN KEY([OrderID])
REFERENCES [Orders] ([OrderID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Products] FOREIGN KEY([ProductID])
REFERENCES [Products] ([ProductID])
```

Bazy Danych – kartkówka zaliczeniowa – 19.12.2011 Imię, nazwisko:

Z podanych zadań wybierz i rozwiąż maksymalnie 5 zadań. Każde zadanie warte jest 10 punktów.

Punktacja zadania:

- 0 punktów – brak rozwiązania lub rozwiązanie całkowicie niepoprawne i znaczne błędy składniowe świadczące o nieznajomości języka SQL
- 3 punkty – rozwiązanie częściowo poprawne i liczne błędy składniowe
- 6 punktów – rozwiązanie częściowo poprawne i brak błędów składniowych lub rozwiązanie poprawne ale liczne błędy składniowe
- 9 punktów – rozwiązanie poprawne i nieliczne błędy składniowe
- 10 punktów – rozwiązanie poprawne, składnia poprawna.

1. Napisz zapytanie, które wybierze z bazy danych Northwind trzy zamówienia o największym łącznym koszcie zamówionych towarów i wypisze kto złożył te zamówienia (CompanyName z tabeli Customers), kiedy zostało złożone, ile zamówiono towarów (suma pól Quantity) i na jaką łączną kwotę (Quantity*UnitPrice).

Hint: wykorzystaj tabele: Customers, Orders oraz [Order Details]

2. Napisz zapytanie, które z tabeli produktów wybierze te, których cena jest niższa niż średnia cen wszystkich produktów i przepisze je do nowej tabeli o nazwie CheapProducts.

Hint: wykorzystaj tabele: Products

3. Stwórz tabelę Struktura zawierającą pola: id, nazwisko (varchar), stopien (varchar), idPrzelozonego. Id jest kluczem głównym, nazwisko i stopien nie mogą być puste, stopień musi być jednym z napisów z listy („PVT”, „PV2”, „PFC”, „SPC”, „CPL”, „SGT”, „SSG”, „SFC”, „MSG”, „ISG”, „SGM”, „CSM”, „SMA”), idPrzelozonego może być NULL, jest kluczem obcym do tej samej tabeli (Struktura).

Wstaw do tabeli następujące dane:

```
(1, Smith, „SGM”, NULL)
(2, Celko, „MSG”, NULL)
(3, Ramirez, „SFC”, 2)
(4, Kovalsky, „SSG”, 1)
(5, Plateu, „SGT”, 1)
(6, Gore, „CPL”, 5)
```

Dla każdej osoby z tabeli Employees z bazy danych Northwind wstaw jej nazwisko (LastName) jako nazwisko, stopien jako „PVT” oraz przelozonego o id 3 dla osób których nazwisko kończy się na „n”, a 4 dla pozostałych osób.

4. Dla tabeli Struktura jak w zadaniu 3 napisz zapytanie (używając CTE), które zwróci wszystkie osoby które są podwładnymi (bezpośrednimi i pośrednimi) osoby o nazwisku „Smith”.
5. Zaproponuj rozwiązanie pamiętające zmiany w stopniach dla osób z tabeli Schemat z zadania 3. Niech będzie to nowa tabela, która będzie pamiętać, kto awansował (idOsoby) na jaki stopień (stopien), kiedy (dataOtrzymania). Usuń kolumnę stopien z tabeli Schemat. Stwórz widok, który zwraca wszystkie osoby, wraz z aktualnie posiadanym stopniem.
6. Załóżmy że, mamy daną tabelę pamiętającą awans zawodowy osób jak w zadaniu 5. Stwórz funkcję, która dla podanej jako parametr osoby (poprzez id osoby) zwraca jej aktualny stopień (jeśli wykonałeś zadanie 5 możesz założyć, że stworzony tam widok jest dostępny).

Tabele bazy danych Northwind:

```
CREATE TABLE [Customers](
    [CustomerID] [nchar](5) NOT NULL PRIMARY KEY,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
)
CREATE TABLE [Employees](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [LastName] [nvarchar](20) NOT NULL,
    [FirstName] [nvarchar](10) NOT NULL,
    [Title] [nvarchar](30) NULL,
    [TitleOfCourtesy] [nvarchar](25) NULL,
    [BirthDate] [datetime] NULL,
    [HireDate] [datetime] NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [HomePhone] [nvarchar](24) NULL,
    [Extension] [nvarchar](4) NULL,
    [Photo] [image] NULL,
    [Notes] [ntext] NULL,
    [ReportsTo] [int] NULL,
    [PhotoPath] [nvarchar](255) NULL,
)
CREATE TABLE [Orders](
    [OrderID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [CustomerID] [nchar](5) NULL,
    [EmployeeID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [RequiredDate] [datetime] NULL,
    [ShippedDate] [datetime] NULL,
    [ShipVia] [int] NULL,
    [Freight] [money] NULL,
    [ShipName] [nvarchar](40) NULL,
    [ShipAddress] [nvarchar](60) NULL,
    [ShipCity] [nvarchar](15) NULL,
    [ShipRegion] [nvarchar](15) NULL,
    [ShipPostalCode] [nvarchar](10) NULL,
    [ShipCountry] [nvarchar](15) NULL,
)
CREATE TABLE [Products](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](40) NOT NULL,
    [SupplierID] [int] NULL,
    [CategoryID] [int] NULL,
    [QuantityPerUnit] [nvarchar](20) NULL,
    [UnitPrice] [money] NULL,
    [UnitsInStock] [smallint] NULL,
    [UnitsOnOrder] [smallint] NULL,
    [ReorderLevel] [smallint] NULL,
    [Discontinued] [bit] NOT NULL,
)
CREATE TABLE [Order Details](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [Quantity] [smallint] NOT NULL,
    [Discount] [real] NOT NULL,
)
ALTER TABLE [Employees] ADD CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Customers] FOREIGN KEY([CustomerID])
REFERENCES [Customers] ([CustomerID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Employees] FOREIGN KEY([EmployeeID])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Orders] FOREIGN KEY([OrderID])
REFERENCES [Orders] ([OrderID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Products] FOREIGN KEY([ProductID])
REFERENCES [Products] ([ProductID])
```

Bazy Danych – kartkówka zaliczeniowa – 19.12.2011 Imię, nazwisko:

Z podanych zadań wybierz i rozwiąż maksymalnie 5 zadań. Każde zadanie warte jest 10 punktów.

Punktacja zadania:

- 0 punktów – brak rozwiązania lub rozwiązanie całkowicie niepoprawne i znaczne błędy składniowe świadczące o nieznajomości języka SQL
- 3 punkty – rozwiązanie częściowo poprawne i liczne błędy składniowe
- 6 punktów – rozwiązanie częściowo poprawne i brak błędów składniowych lub rozwiązanie poprawne ale liczne błędy składniowe
- 9 punktów – rozwiązanie poprawne i nieliczne błędy składniowe
- 10 punktów – rozwiązanie poprawne, składnia poprawna.

1. Napisz zapytanie, które wybierze z bazy danych Northwind trzy zamówienia o największym łącznym koszcie zamówionych towarów i wypisze kto złożył te zamówienia (CompanyName z tabeli Customers), kiedy zostało złożone, ile zamówiono towarów (suma pól Quantity) i na jaką łączną kwotę (Quantity*UnitPrice).

Hint: wykorzystaj tabele: Customers, Orders oraz [Order Details]

2. Napisz zapytanie, które z tabeli produktów wybierze te, których cena jest niższa niż średnia cen wszystkich produktów i przepisze je do nowej tabeli o nazwie CheapProducts.

Hint: wykorzystaj tabele: Products

3. Stwórz tabelę Struktura zawierającą pola: id, nazwisko (varchar), stopien (varchar), idPrzelozonego. Id jest kluczem głównym, nazwisko i stopien nie mogą być puste, stopień musi być jednym z napisów z listy („PVT”, „PV2”, „PFC”, „SPC”, „CPL”, „SGT”, „SSG”, „SFC”, „MSG”, „ISG”, „SGM”, „CSM”, „SMA”), idPrzelozonego może być NULL, jest kluczem obcym do tej samej tabeli (Struktura).

Wstaw do tabeli następujące dane:

```
(1, Smith, „SGM”, NULL)
(2, Celko, „MSG”, NULL)
(3, Ramirez, „SFC”, 2)
(4, Kovalsky, „SSG”, 1)
(5, Plateu, „SGT”, 1)
(6, Gore, „CPL”, 5)
```

Dla każdej osoby z tabeli Employees z bazy danych Northwind wstaw jej nazwisko (LastName) jako nazwisko, stopien jako „PVT” oraz przelozonego o id 3 dla osób których nazwisko kończy się na „n”, a 4 dla pozostałych osób.

4. Dla tabeli Struktura jak w zadaniu 3 napisz zapytanie (używając CTE), które zwróci wszystkie osoby które są podwładnymi (bezpośrednimi i pośrednimi) osoby o nazwisku „Smith”.
5. Zaproponuj rozwiązanie pamiętające zmiany w stopniach dla osób z tabeli Schemat z zadania 3. Niech będzie to nowa tabela, która będzie pamiętać, kto awansował (idOsoby) na jaki stopień (stopien), kiedy (dataOtrzymania). Usuń kolumnę stopien z tabeli Schemat. Stwórz widok, który zwraca wszystkie osoby, wraz z aktualnie posiadanym stopniem.
6. Załóżmy że, mamy daną tabelę pamiętającą awans zawodowy osób jak w zadaniu 5. Stwórz funkcję, która dla podanej jako parametr osoby (poprzez id osoby) zwraca jej aktualny stopień (jeśli wykonałeś zadanie 5 możesz założyć, że stworzony tam widok jest dostępny).

Tabele bazy danych Northwind:

```
CREATE TABLE [Customers](
    [CustomerID] [nchar](5) NOT NULL PRIMARY KEY,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
)
CREATE TABLE [Employees](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [LastName] [nvarchar](20) NOT NULL,
    [FirstName] [nvarchar](10) NOT NULL,
    [Title] [nvarchar](30) NULL,
    [TitleOfCourtesy] [nvarchar](25) NULL,
    [BirthDate] [datetime] NULL,
    [HireDate] [datetime] NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [HomePhone] [nvarchar](24) NULL,
    [Extension] [nvarchar](4) NULL,
    [Photo] [image] NULL,
    [Notes] [ntext] NULL,
    [ReportsTo] [int] NULL,
    [PhotoPath] [nvarchar](255) NULL,
)
CREATE TABLE [Orders](
    [OrderID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [CustomerID] [nchar](5) NULL,
    [EmployeeID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [RequiredDate] [datetime] NULL,
    [ShippedDate] [datetime] NULL,
    [ShipVia] [int] NULL,
    [Freight] [money] NULL,
    [ShipName] [nvarchar](40) NULL,
    [ShipAddress] [nvarchar](60) NULL,
    [ShipCity] [nvarchar](15) NULL,
    [ShipRegion] [nvarchar](15) NULL,
    [ShipPostalCode] [nvarchar](10) NULL,
    [ShipCountry] [nvarchar](15) NULL,
)
CREATE TABLE [Products](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](40) NOT NULL,
    [SupplierID] [int] NULL,
    [CategoryID] [int] NULL,
    [QuantityPerUnit] [nvarchar](20) NULL,
    [UnitPrice] [money] NULL,
    [UnitsInStock] [smallint] NULL,
    [UnitsOnOrder] [smallint] NULL,
    [ReorderLevel] [smallint] NULL,
    [Discontinued] [bit] NOT NULL,
)
CREATE TABLE [Order Details](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [Quantity] [smallint] NOT NULL,
    [Discount] [real] NOT NULL,
)
ALTER TABLE [Employees] ADD CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Customers] FOREIGN KEY([CustomerID])
REFERENCES [Customers] ([CustomerID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Employees] FOREIGN KEY([EmployeeID])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Orders] FOREIGN KEY([OrderID])
REFERENCES [Orders] ([OrderID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Products] FOREIGN KEY([ProductID])
REFERENCES [Products] ([ProductID])
```


Bazy Danych – kartkówka zaliczeniowa – 19.12.2011 Imię, nazwisko:

Z podanych zadań wybierz i rozwiąż maksymalnie 5 zadań. Każde zadanie warte jest 10 punktów.

Punktacja zadania:

- 0 punktów – brak rozwiązania lub rozwiązanie całkowicie niepoprawne i znaczne błędy składniowe świadczące o nieznajomości języka SQL
- 3 punkty – rozwiązanie częściowo poprawne i liczne błędy składniowe
- 6 punktów – rozwiązanie częściowo poprawne i brak błędów składniowych lub rozwiązanie poprawne ale liczne błędy składniowe
- 9 punktów – rozwiązanie poprawne i nieliczne błędy składniowe
- 10 punktów – rozwiązanie poprawne, składnia poprawna.

1. Napisz zapytanie, które wybierze z bazy danych Northwind trzy zamówienia o największym łącznym koszcie zamówionych towarów i wypisze kto złożył te zamówienia (CompanyName z tabeli Customers), kiedy zostało złożone, ile zamówiono towarów (suma pól Quantity) i na jaką łączną kwotę (Quantity*UnitPrice).

Hint: wykorzystaj tabele: Customers, Orders oraz [Order Details]

2. Napisz zapytanie, które z tabeli produktów wybierze te, których cena jest niższa niż średnia cen wszystkich produktów i przepisze je do nowej tabeli o nazwie CheapProducts.

Hint: wykorzystaj tabele: Products

3. Stwórz tabelę Struktura zawierającą pola: id, nazwisko (varchar), stopien (varchar), idPrzelozonego. Id jest kluczem głównym, nazwisko i stopien nie mogą być puste, stopień musi być jednym z napisów z listy („PVT”, „PV2”, „PFC”, „SPC”, „CPL”, „SGT”, „SSG”, „SFC”, „MSG”, „ISG”, „SGM”, „CSM”, „SMA”), idPrzelozonego może być NULL, jest kluczem obcym do tej samej tabeli (Struktura).

Wstaw do tabeli następujące dane:

```
(1, Smith, „SGM”, NULL)
(2, Celko, „MSG”, NULL)
(3, Ramirez, „SFC”, 2)
(4, Kovalsky, „SSG”, 1)
(5, Plateu, „SGT”, 1)
(6, Gore, „CPL”, 5)
```

Dla każdej osoby z tabeli Employees z bazy danych Northwind wstaw jej nazwisko (LastName) jako nazwisko, stopien jako „PVT” oraz przelozonego o id 3 dla osób których nazwisko kończy się na „n”, a 4 dla pozostałych osób.

4. Dla tabeli Struktura jak w zadaniu 3 napisz zapytanie (używając CTE), które zwróci wszystkie osoby które są podwładnymi (bezpośrednimi i pośrednimi) osoby o nazwisku „Smith”.
5. Zaproponuj rozwiązanie pamiętające zmiany w stopniach dla osób z tabeli Schemat z zadania 3. Niech będzie to nowa tabela, która będzie pamiętać, kto awansował (idOsoby) na jaki stopień (stopien), kiedy (dataOtrzymania). Usuń kolumnę stopien z tabeli Schemat. Stwórz widok, który zwraca wszystkie osoby, wraz z aktualnie posiadanym stopniem.
6. Załóżmy że, mamy daną tabelę pamiętającą awans zawodowy osób jak w zadaniu 5. Stwórz funkcję, która dla podanej jako parametr osoby (poprzez id osoby) zwraca jej aktualny stopień (jeśli wykonałeś zadanie 5 możesz założyć, że stworzony tam widok jest dostępny).

Tabele bazy danych Northwind:

```
CREATE TABLE [Customers](
    [CustomerID] [nchar](5) NOT NULL PRIMARY KEY,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
)
CREATE TABLE [Employees](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [LastName] [nvarchar](20) NOT NULL,
    [FirstName] [nvarchar](10) NOT NULL,
    [Title] [nvarchar](30) NULL,
    [TitleOfCourtesy] [nvarchar](25) NULL,
    [BirthDate] [datetime] NULL,
    [HireDate] [datetime] NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [HomePhone] [nvarchar](24) NULL,
    [Extension] [nvarchar](4) NULL,
    [Photo] [image] NULL,
    [Notes] [ntext] NULL,
    [ReportsTo] [int] NULL,
    [PhotoPath] [nvarchar](255) NULL,
)
CREATE TABLE [Orders](
    [OrderID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [CustomerID] [nchar](5) NULL,
    [EmployeeID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [RequiredDate] [datetime] NULL,
    [ShippedDate] [datetime] NULL,
    [ShipVia] [int] NULL,
    [Freight] [money] NULL,
    [ShipName] [nvarchar](40) NULL,
    [ShipAddress] [nvarchar](60) NULL,
    [ShipCity] [nvarchar](15) NULL,
    [ShipRegion] [nvarchar](15) NULL,
    [ShipPostalCode] [nvarchar](10) NULL,
    [ShipCountry] [nvarchar](15) NULL,
)
CREATE TABLE [Products](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](40) NOT NULL,
    [SupplierID] [int] NULL,
    [CategoryID] [int] NULL,
    [QuantityPerUnit] [nvarchar](20) NULL,
    [UnitPrice] [money] NULL,
    [UnitsInStock] [smallint] NULL,
    [UnitsOnOrder] [smallint] NULL,
    [ReorderLevel] [smallint] NULL,
    [Discontinued] [bit] NOT NULL,
)
CREATE TABLE [Order Details](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [Quantity] [smallint] NOT NULL,
    [Discount] [real] NOT NULL,
)
ALTER TABLE [Employees] ADD CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Customers] FOREIGN KEY([CustomerID])
REFERENCES [Customers] ([CustomerID])
ALTER TABLE [Orders] ADD CONSTRAINT [FK_Orders_Employees] FOREIGN KEY([EmployeeID])
REFERENCES [Employees] ([EmployeeID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Orders] FOREIGN KEY([OrderID])
REFERENCES [Orders] ([OrderID])
ALTER TABLE [Order Details] ADD CONSTRAINT [FK_Order_Details_Products] FOREIGN KEY([ProductID])
REFERENCES [Products] ([ProductID])
```