

Cvičení 9: Poddotazy

- 1) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří zatím neuskutečnili žádnou objednávku.
- 2) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří si někdy objednali více než 100 kusů jednoho produktu v rámci jedné objednávky.
- 3) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří mají aktuálně nějakou nezaplacenou objednávku (*OrderStatus.OrderStatus_Name* = 'Nezaplacená').
- 4) Napište dotaz, který vrátí data agregovaná na úrovni zákazníka a produktu z tabulky Order, a to konkrétně celkové množství daného produktu, které si zákazník kdy objednal (*PočetObjednanýchKusů*) a počet zákaznickových objednávek, ve kterých se daný produkt vyskytoval (*PočetObjednávekDanéhoProduktu*). Nakonec výsledek dotazu seřadíme **sestupně** dle **počtu objednaných kusů**.
- 5) Nyní využijte dotaz z minulé úlohy k tomu, abyste vypočítali průměrný počet kusů jednoho produktu, který si zákazník objedná v rámci všech svých objednávek (*PrůměrnýPočetObjednanýchKusů*). Vytvořený dotaz by měl k tomu také vrátit i **průměrný počet objednávek** (*Průměrný počet objednávek*), v rámci kterých zákazníci daný počet konkrétního produktu nakoupí.
- 6) Napište dotaz, který vrátí seznam objednávek s daným množstvím objednaného produktu a jednotkovou cenou tohoto produktu. Navíc vytvořte nový sloupec s názvem *VýšeObjednávky*, který se vypočítá vynásobením množství a jednotkové ceny.
- 7) Do dotazu z předchozí úlohy přidejte sloupec s názvem *PrůměrVýšeObjednávky*, který odpovídá průměrné hodnotě vypočítané ze všech objednávek. Tato hodnota tedy bude pro všechny řádky výsledné tabulky stejná.

- 8) Do dotazu z předchozí úlohy přidejte sloupec s názvem *RozdílVýšeObjednávky*, který bude odpovídat rozdílu výše konkrétní objednávky a průměrné hodnoty ze všech objednávek.
- 9) Napište dotaz, který vrátí jméno a váhu tří nejtěžších produktů z tabulky **Product**. Navíc do dotazu přidejte sloupec **Popis**, který bude plněný hodnotou 'Nejtěžší produkty' a ve výsledné tabulce bude na první místě.
- 10) Napište dotaz, který vrátí jméno a váhu tří nejlehčích produktů z tabulky **Product**. Navíc do dotazu přidejte sloupec **Popis**, který bude plněný hodnotou 'Nejlehčí produkty' a ve výsledné tabulce bude na první místě.
- 11) Dotazy z úloh 9 a 10 spojte nějakým množinovým operátorem tak, aby vznikla jedna tabulka obsahující jak 3 nejtěžší, tak i 3 nejlehčí produkty s příslušným popisem.

Poznámka: Při práci s množinovými operátory lze používat klauzuli **ORDER BY** uvnitř jednotlivých dotazů pouze případě, že je celý dotaz s **ORDER BY** vložen do poddotazu – **subselectu** (a z pohledu SQL je z něj tedy udělána samostatná tabulka).

Cvičení 9: Poddotazy

- 1) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří zatím neuskutečnili žádnou objednávku.

Použijeme víceřádkový poddotaz, který vrací seznam ID zákazníků, kteří se vyskytují v tabulce **Order**, a tedy již někdy uskutečnili nějakou objednávku.

Poté použijeme **WHERE** podmínku s operátorem **NOT IN**, díky které získáme jen záznamy pro ID zákazníků **nepatřící** do skupiny ID v množině poddotazu.

```
SELECT Customer_ID
       ,Customer_Name
       ,Customer_Surname
       ,Date_Birth
FROM [csv].[Customer]
WHERE Customer_ID NOT IN
      (SELECT Customer_ID FROM [csv].[Order])
```

- 2) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří si někdy objednali více než 100 kusů jednoho produktu v rámci jedné objednávky.

Použijeme obdobný dotaz jako v minulé úloze, jen do něj přidáme **WHERE** podmínku na **množství větší než 100**.

```
SELECT Customer_ID
       ,Customer_Name
       ,Customer_Surname
       ,Date_Birth
FROM [csv].[Customer]
WHERE Customer_ID IN
      (SELECT Customer_ID
       FROM [csv].[Order]
       WHERE Quantity > 100
      )
```

- 3) Napište dotaz, který vrátí ID, jméno, příjmení a datum narození zákazníků, kteří mají aktuálně nějakou nezaplacenou objednávku (*OrderStatus.OrderStatus_Name = 'Nezaplacená'*).

V tomto případě je potřeba ke zdrojové tabulce **Order** v poddotazu připojit tabulku **OrderStatus**, na základě jejíhož atributu má probíhat filtrování dat pomocí **WHERE** podmínky. Nesmíme však zapomenout, že v **SELECT** klauzuli nesmí být více než jeden sloupec, protože výsledek poddotazu musí odpovídat **množině hodnot sloupce Customer_ID**, se kterými SQL poté porovnává každé **Customer.Customer_ID**.

```
SELECT Customer_ID
      ,Customer_Name
      ,Customer_Surname
      ,Date_Birth
FROM [csv].[Customer]
WHERE Customer_ID IN
(
  SELECT ord.Customer_ID
  FROM [csv].[Order] ord
  LEFT JOIN [dbo].[OrderStatus] stat
  ON ord.Order_Status = stat.OrderStatus_ID
  WHERE stat.OrderStatus_Name = 'Nezaplacená'
)
```

- 4) Napište dotaz, který vrátí data agregovaná na úrovni zákazníka a produktu z tabulky Order, a to konkrétně celkové množství daného produktu, které si zákazník kdy objednal (*PočetObjednanýchKusů*) a počet zákaznickových objednávek, ve kterých se daný produkt vyskytoval (*PočetObjednávekDanéhoProduktu*). Nakonec výsledek dotazu seřadíme **sestupně** dle **počtu objednaných kusů**.

Z tabulky **Order** budeme chtít vrátit sloupce **Customer_ID** a **Product_ID**, na základě nichž data také seskupíme, tzn. tyto atributy dáme do klauzule **GROUP BY**. Následně vytvoříme dva kalkulované sloupce s agregačními funkcemi **SUM** a **COUNT**, jelikož v jednom případě **sčítáme množství** ze všech objednávek, které obsahují danou kombinaci **zákazník – produkt** a v tom druhém jen **zjišťujeme počet řádků**, na kterých se tato kombinace vyskytuje.

```
SELECT Customer_ID
      ,Product_ID
      ,SUM(Quantity) AS PočetObjednanýchKusů
      ,COUNT(Product_ID) AS PočetObjednávekDanéhoProduktu
FROM [csv].[Order]
GROUP BY Customer_ID, Product_ID
ORDER BY PočetObjednanýchKusů DESC
```

Výsledná tabulka vypadá následovně:

	Customer_ID	Product_ID	PočetObjednanýchKusů	PočetObjednávekDanéhoProduktu
1	C204	P115	460	1
2	C104	P110	295	3
3	C180	P119	293	3
4	C190	P101	284	3
5	C120	P112	270	1
6	C200	P115	253	6
7	C165	P119	251	4
8	C104	P106	240	2

Pro kontrolu, že jsou výpočty v dotazu prováděny správně, si můžeme vybrat konkrétní záznam a najít pro něj detailní informace v tabulce **Order**.

```
SELECT * FROM [csv].[Order]
WHERE Customer_ID = 'C104' AND Product_ID = 'P110'
```

	Order_ID	Order_Date	Customer_ID	Product_ID	Quantity	Order_Status	Currency
1	O1141	2021-01-02	C104	P110	289	1	CZK
2	O1153	2020-04-08	C104	P110	3	1	CZK
3	O2659	2021-01-22	C104	P110	3	1	CZK

- 5) Nyní využijte dotaz z minulé úlohy k tomu, abyste vypočítali průměrný počet kusů jednoho produktu, který si zákazník objedná v rámci všech svých objednávek (*PrůměrnýPočetObjednanýchKusů*). Vytvořený dotaz by měl k tomu také vrátit i **průměrný počet objednávek** (*Průměrný počet objednávek*), v rámci kterých zákazníci daný počet konkrétního produktu nakoupí.

Dotaz z minulé úlohy použijeme jako **zdrojovou tabulku** nového dotazu. Dáme jej tedy do závorek za klauzuli **FROM** a vytvoříme pro něj **alias** (např. písmeno a), jelikož každá tabulka v SQL musí mít nějaký název. Jelikož v rámci **subselectu** nelze používat klauzuli **ORDER BY**, pokud v něm zároveň není definována klauzule **TOP** nebo **OFFSET**, můžeme tu klauzuli vymazat nebo „zakomentovat“.

```
SELECT AVG(PočetObjednanýchKusů) AS PrůměrnýPočetObjednanýchKusů
      ,AVG(PočetObjednávekDanéhoProduktu) AS PrůměrnýPočetObjednávek
FROM
```

```
(
SELECT Customer_ID
      ,Product_ID
      ,SUM(Quantity) AS PočetObjednanýchKusů
      ,COUNT(Product_ID) AS PočetObjednávekDanéhoProduktu
FROM [csv].[Order]
GROUP BY Customer_ID, Product_ID
--ORDER BY PočetObjednanýchKusů DESC
```

```
) a
```

Poté v rámci klauzule **SELECT** definujeme dva nově vytvářené sloupce, jejichž hodnotu vypočítáme **agregační funkcí AVG**, jelikož se v obou případech jedná o **průměr** hodnot vypočítaných poddotazem.

Z výsledku celého dotazu poté můžeme zjistit, že **průměrně** zákazník nakoupí 8 kusů od každého produktu v rámci dvou objednávek.

	PrůměrnýPočetObjednanýchKusů	PrůměrnýPočetObjednávek
1	8	2

Jelikož se však jedná o **jednoduchý výpočet průměrných hodnot**, stálo by za to zjistit, zda nám některou z hodnot výrazně neovlivňuje nějaký „**outlier**“, tedy příliš **vysoká** či **nízká** hodnota, která ani řádově neodpovídá distribuci zbylých hodnot.

Kdybychom např. manuálně odfiltrovali 5 zákazníků, kteří ve výsledku dotazu ze čtvrté úlohy mají nejvyšší **počet objednaných kusů** určitého produktu (výrazně vyšší než většina zbylých záznamů), vyšel by **průměrný počet objednaných kusů** rázem o jeden menší.

	Customer_ID	Product_ID	PočetObjednanýchKusů	PočetObjednávekDanéhoProduktu
1	C204	P115	460	1
2	C104	P110	295	3
3	C180	P119	293	3
4	C190	P101	284	3
5	C120	P112	270	1
6	C200	P115	253	6
7	C165	P119	251	4
8	C104	P106	240	2

```
SELECT AVG(PočetObjednanýchKusů) AS PrůměrnýPočetObjednanýchKusů
      ,AVG(PočetObjednávekDanéhoProduktu) AS PrůměrnýPočetObjednávek
FROM
(
  SELECT Customer_ID
        ,Product_ID
        ,SUM(Quantity) AS PočetObjednanýchKusů
        ,COUNT(Product_ID) AS PočetObjednávekDanéhoProduktu
  FROM [csv].[Order]
  WHERE Customer_ID NOT IN ('C204', 'C104', 'C180', 'C190', 'C120')
  GROUP BY Customer_ID, Product_ID
  --ORDER BY PočetObjednanýchKusů DESC
) a
```

	PrůměrnýPočetObjednanýchKusů	PrůměrnýPočetObjednávek
1	7	2

- 6) Napište dotaz, který vrátí seznam objednávek s daným množstvím objednaného produktu a jednotkovou cenou tohoto produktu. Navíc vytvořte nový sloupec s názvem **VýšeObjednávky**, který se vypočítá vynásobením množství a jednotkové ceny.

```
SELECT ord.Order_ID
      ,ord.Quantity
      ,prod.Unit_Price
      ,ord.Quantity * prod.Unit_Price AS VýšeObjednávky
FROM [csv].[Order] ord
LEFT JOIN [csv].[Product] prod
ON ord.Product_ID = prod.Product_ID
```

- 7) Do dotazu z předchozí úlohy přidejte sloupec s názvem **PrůměrVýšeObjednávky**, který odpovídá průměrné hodnotě vypočítané ze všech objednávek. Tato hodnota tedy bude pro všechny řádky výsledné tabulky stejná.

Nejdříve si můžeme v samostatném dotazu vypočítat požadovaný průměr hodnot všech objednávek. Pro zjednodušení nebudeme brát v potaz možné měnové rozdíly.

```
SELECT AVG(ord.Quantity * prod.Unit_Price) AS PrůměrVýšeObjednávky
FROM [csv].[Order] ord
LEFT JOIN [csv].[Product] prod
ON ord.Product_ID = prod.Product_ID
```

	PrůměrVýšeObjednávky
1	3925

Tuto hodnotu budeme poté chtít vložit do dotazu z minulé úlohy. Dáme tedy celý výše vytvořený dotaz do závorek a použijeme jako poddotaz definující hodnotu nového sloupce **PrůměrVýšeObjednávky**.

```
SELECT ord.Order_ID
      ,ord.Quantity
      ,prod.Unit_Price
      ,ord.Quantity * prod.Unit_Price AS VýšeObjednávky
      ,(SELECT AVG(ord.Quantity * prod.Unit_Price)
        FROM [csv].[Order] ord
        LEFT JOIN [csv].[Product] prod
        ON ord.Product_ID = prod.Product_ID) AS PrůměrVýšeObjednávky
FROM [csv].[Order] ord
LEFT JOIN [csv].[Product] prod
ON ord.Product_ID = prod.Product_ID
```

	Order_ID	Quantity	Unit_Price	VýšeObjednávky	PrůměrVýšeObjednávky
1	O1000	2	1000	2000	3925
2	O1001	1	160	160	3925
3	O1002	3	1700	5100	3925
4	O1003	1	1800	1800	3925
5	O1004	4	1200	4800	3925

- 8) Do dotazu z předchozí úlohy přidejte sloupec s názvem *RozdílVýšeObjednávky*, který bude odpovídat rozdílu výše konkrétní objednávky a průměrné hodnoty ze všech objednávek.

Dotaz z předchozí úlohy bude rozšířen o jeden sloupec. Jeho hodnoty vypočítáme jako rozdíl hodnot sloupců **VýšeObjednávky** a **PrůměrVýšeObjednávky**. V rámci **SELECT** klauzule se však nelze odkazovat na vytvořené **aliasy**, a proto pro výpočet hodnot nového sloupce musíme použít původní výpočty každého ze zmíněných sloupců.

```
SELECT ord.Order_ID
      ,ord.Quantity
      ,prod.Unit_Price
      ,ord.Quantity * prod.Unit_Price AS VýšeObjednávky
      ,(SELECT AVG(ord.Quantity * prod.Unit_Price)
        FROM [csv].[Order] ord
        LEFT JOIN [csv].[Product] prod
        ON ord.Product_ID = prod.Product_ID) AS PrůměrVýšeObjednávky
      ,(ord.Quantity * prod.Unit_Price) -
      (SELECT AVG(ord.Quantity * prod.Unit_Price)
        FROM [csv].[Order] ord
        LEFT JOIN [csv].[Product] prod
        ON ord.Product_ID = prod.Product_ID) AS RozdílVýšeObjednávky
FROM [csv].[Order] ord
LEFT JOIN [csv].[Product] prod
ON ord.Product_ID = prod.Product_ID
```

Díky tomu jsme poté u každé objednávky schopni říct, zda je její hodnota nadprůměrná či podprůměrná, a o jakou částku se od tohoto průměru liší.

	Order_ID	Quantity	Unit_Price	VýšeObjednávky	PrůměrVýšeObjednávky	RozdílVýšeObjednávky
1	O1000	2	1000	2000	3925	-1925
2	O1001	1	160	160	3925	-3765
3	O1002	3	1700	5100	3925	1175
4	O1003	1	1800	1800	3925	-2125
5	O1004	4	1200	4800	3925	875
6	O1005	2	1000	2000	3925	-1925

- 9) Napište dotaz, který vrátí jméno a váhu tří nejtěžších produktů z tabulky **Product**. Navíc do dotazu přidejte sloupec **Popis**, který bude plněný hodnotou **'Nejtěžší produkty'** a ve výsledné tabulce bude na první místě.

```
SELECT TOP(3) 'Nejtěžší produkty' AS Popis
              ,Product_Name
              ,Weight
FROM [csv].[Product]
ORDER BY Weight DESC
```


- 10) Napište dotaz, který vrátí jméno a váhu tří nejlehčích produktů z tabulky **Product**. Navíc do dotazu přidejte sloupec **Popis**, který bude plněný hodnotou 'Nejlehčí produkty' a ve výsledné tabulce bude na první místě.

```
SELECT TOP(3) 'Nejlehčí produkty' AS Popis
              ,Product_Name
              ,Weight
FROM [csv].[Product]
ORDER BY Weight
```

- 11) Dotazy z úloh 9 a 10 spojte nějakým množinovým operátorem tak, aby vznikla jedna tabulka obsahující jak 3 nejtěžší, tak i 3 nejlehčí produkty s příslušným popisem.

Poznámka: Při práci s množinovými operátory lze používat klauzuli **ORDER BY** uvnitř jednotlivých dotazů pouze případě, že je celý dotaz s **ORDER BY** vložen do poddotazu – **subselectu** (a z pohledu SQL je z něj tedy udělána samostatná tabulka).

Z obou dotazů je postupně nutné udělat poddotazy, aby mohla být využita klauzule **ORDER BY**, která je potřebná pro správné nalezení **TOP** hodnot. Začneme tedy prvním dotazem, který nejprve dáme do **závorek** a tuto závorku pojmenujeme (např. písmenem **a**).

Tím si umožníme se na tento dotaz odkazovat jako na **zdrojovou tabulku** pomocí výrazu **SELECT * FROM**. Získáme tak úplně stejná data, jako když jsme dotaz pustili samotný, ale navíc je možné jej použít v rámci množinového operátoru **UNION**.

Poté tu samou úpravu uděláme i pro druhý dotaz.

```
SELECT * FROM
  (SELECT TOP(3) 'Nejtěžší produkty' AS Popis
    ,Product_Name
    ,Weight
  FROM [csv].[Product]
  ORDER BY Weight DESC) a
```

UNION

```
SELECT * FROM
  (SELECT TOP(3) 'Nejlehčí produkty' AS Popis
    ,Product_Name
    ,Weight
  FROM [csv].[Product]
  ORDER BY Weight) b
```

	Popis	Product_Name	Weight
1	Nejlehčí produkty	Lego	188.40
2	Nejlehčí produkty	Shirt	315.00
3	Nejlehčí produkty	T-Shirt	243.00
4	Nejtěžší produkty	Cupboard	1932.00
5	Nejtěžší produkty	Sofa	3432.00
6	Nejtěžší produkty	Table	2072.00