

Cvičení 6: Klauzule **DISTINCT**, **GROUP BY**, **HAVING** a **PARTITION BY**

- 1) **Naimportujte zdrojovou tabulku `Order_CSV` do vaší databáze v SSMS. Měla by přitom být splněna následující kritéria:**
 - Tabulka bude pojmenována jako **Order** a uložena ve **schématu csv**.
 - **Primárním klíčem** je sloupec **Order_ID**, který je složen vždy z 5 znaků.
 - **Kód produktu** a **zákazníka** musí mít stejný datový typ jako v tabulkách **Product** a **Customer**.
 - Veškeré **datumové sloupce** by v databázi měly být uloženy jako datum bez času.
 - **Kód měny** objednávky je vždy třípísmenný textový řetězec.
 - Sloupec **Quantity**, tedy množství položek na objednávce, se běžně pohybuje v řádu jednotek, maximálně stovek (do cca 500).
 - **Status objednávky** je vždy definován jako jednociferný kód (např. 1, 2, 3).
- 2) **Nalezněte abecedně seřazený seznam různých křestních jmen, která se vyskytují v tabulce Customer.** Poté z nich vyberte pouze ta **mužská**.
- 3) **Pozor, v seznamu z předchozí úlohy se objeví i dvě jména, která by tam být neměla.** Podívejte se na tyto záznamy ve větším detailu. Následně je v databázi pomocí výrazu **UPDATE** aktualizujte tak, aby se již příště objevily ve správné kategorii dle pohlaví zákazníka.
- 4) **Nalezněte různé hodnoty vyskytující se ve sloupci `Order.Order_Status` a k nim množství objednávek s daným statusem. Záznamy seřadte dle hodnot sloupce `Order.Order_Status`.**
- 5) **Na základě hodnot sloupce `Order.Order_Status` vytvořte číselníkovou tabulku `OrderStatus`, která bude obsahovat sloupce `OrderStatus_ID` (se stejným datovým typem jako `Order.Order_Status`) a `OrderStatus_Name` (textový řetězec s diakritikou a maximálně 20ti znaky). Po vytvoření tabulky do ní vložte záznamy ('1', 'Zaplacená'), ('2', 'Nezaplacená'), ('3', 'Zrušená').**

- 6) V tabulce Order zjistěte, jaká různá množství položek (sloupec *Quantity*) byla zatím zákaznící v rámci jedné objednávky objednána. Seřadte tyto záznamy od nejvyššího množství po to nejnižší.
- 7) Nalezněte seznam TOP 10 zákaznících dle počtu provedených objednávek. Poté nalezněte také TOP 10 těch, kteří si objednali největší množství položek.
- 8) Záznamy v tabulce Order seskupte podle množství objednaných položek tak, abyste zjistili, jaká množství jsou objednána nejčastěji.
- 9) Záznamy v tabulce Order seskupte podle zákaznících a pro každého tak zjistěte:
- počet provedených objednávek
 - celkovou sumu nakoupených položek
 - počet **různých** produktů, které si zákazník kdy objednal
 - datum první objednávky
 - datum poslední objednávky

Zároveň záznamy seřadte **sestupně** podle **počtu provedených objednávek**.

- 10) V tabulce Order nalezněte kombinace zákazníka a konkrétního data, v rámci kterých byly provedeny více než 2 objednávky. Jinými slovy zjistěte, kteří zákazníci a ve kterých dnech provedli více než 2 objednávky. Získané záznamy zároveň **seřadte** tak, aby **nejnovější objednávky** byly v seznamu jako první.
- 11) Napište dotaz, který z tabulky Product vrátí všechny záznamy pro sloupce *Product_Name*, *Product_Category* a *Weight*, a navíc k nim přidá informaci o **nejvyšší váze v rámci příslušné kategorie** (tedy nejvyšší hodnota sloupce *Weight* pro kategorii, do které daný produkt na určitém řádku patří).
- 12) Na základě dotazu z minulé úlohy vypočítejte pro každý záznam tabulky Product rozdíl mezi *maximální cenou z dané produkové kategorie* a *cenou určitého produktu* na daném řádku.
- 13) Napište dotaz, který vrátí všechny záznamy z tabulky Order, přičemž ke každému zákazníkovi přiřadí informaci o jeho celkovém počtu objednávek. Data navíc **seřadte** podle **kódu zákazníka**.

Cvičení 6: Klauzule DISTINCT, GROUP BY, HAVING a PARTITION BY

1) Naimportujte zdrojovou tabulku **Order_CSV** do vaší databáze v SSMS. Měla by přitom být splněna následující kritéria:

- Tabulka bude pojmenována jako **Order** a uložena ve **schématu csv**.
- **Primárním klíčem** je sloupec **Order_ID**, který je složen vždy z 5 znaků.
- **Kód produktu** a **zákazníka** musí mít stejný datový typ jako v tabulkách **Product** a **Customer**.
- Veškeré **datumové sloupce** by v databázi měly být uloženy jako datum bez času.
- **Kód měny** objednávky je vždy třípísmenný textový řetězec.
- Sloupec **Quantity**, tedy množství položek na objednávce, se běžně pohybuje v řádu jednotek, maximálně stovek (do cca 500).
- **Status objednávky** je vždy definován jako jednociferný kód (např. 1, 2, 3).

Import tabulky lze provést zase přes možnost **Tasks – Import Flat File** po kliknutí pravým tlačítkem na název dané databáze. Poté by měly být parametry tabulky nastaveny následujícím způsobem.

The image shows two screenshots of the SQL Server Enterprise Manager interface. The left screenshot shows the 'Specify Input File' wizard with the file path 'C:\Users\Asus\Desktop\Order_CSV.csv', the new table name 'Order', and the table schema 'csv'. The right screenshot shows the 'Modify Columns' wizard with a table schema table.

Column Name	Data Type	Primary Key	Allow Nulls
Order_ID	char(5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Order_Date	date	<input type="checkbox"/>	<input type="checkbox"/>
Customer_ID	char(4)	<input type="checkbox"/>	<input type="checkbox"/>
Product_ID	char(4)	<input type="checkbox"/>	<input type="checkbox"/>
Quantity	smallint	<input type="checkbox"/>	<input type="checkbox"/>
Order_Status	char(1)	<input type="checkbox"/>	<input type="checkbox"/>
Currency	char(3)	<input type="checkbox"/>	<input type="checkbox"/>

2) Nalezněte abecedně seřazený seznam různých křestních jmen, která se vyskytují v tabulce **Customer**. Poté z nich vyberte pouze ta **mužská**.

```
SELECT DISTINCT Customer_Name
FROM [csv].[Customer]
WHERE Gender = 'M'
ORDER BY Customer_Name
```

- 3) **Pozor, v seznamu z předchozí úlohy se objeví i dvě jména, která by tam být neměla.** Podívejte se na tyto záznamy ve větším detailu. Následně je v databázi pomocí výrazu **UPDATE** aktualizujte tak, aby se již příště objevily ve správné kategorii dle pohlaví zákazníka.

Mezi mužskými jmény z předchozího dotazu se vyskytují i jména **Miluše** a **Monika**. Následujícím dotazem tedy najdeme takové záznamy z tabulky **Order**, ve kterých se vyskytuje některé z těchto jmen.

8	Juraj
9	Lukáš
10	Miluše
11	Monika
12	Oldřich

```
SELECT *
FROM [csv].[Customer]
WHERE Customer_Name IN ('Monika', 'Miluše')
```

	Customer_ID	Customer_Name	Customer_Surname	Gender	Date_Birth	Date_Registered
1	C160	Monika	Daňková	M	1985-06-24	2020-11-07 00:00:00.000
2	C162	Miluše	Svobodová	M	1981-01-13	2020-05-19 00:00:00.000
3	C167	Monika	Chrástková	F	1968-06-22	2020-06-05 00:00:00.000

Ve výsledku dotazu vidíme, že u prvních dvou záznamů byly pravděpodobně špatně zadána vstupní data. Pomocí příkazu s klauzulí **UPDATE** je však můžeme opravit.

```
UPDATE [csv].[Customer]
SET Gender = 'F'
WHERE Customer_Name IN ('Monika', 'Miluše')
```

	Customer_ID	Customer_Name	Customer_Surname	Gender	Date_Birth	Date_Registered
1	C160	Monika	Daňková	F	1985-06-24	2020-11-07 00:00:00.000
2	C162	Miluše	Svobodová	F	1981-01-13	2020-05-19 00:00:00.000
3	C167	Monika	Chrástková	F	1968-06-22	2020-06-05 00:00:00.000

Po aktualizaci dat by tedy dotaz z předchozí úlohy již měl vrátit jen mužská jména.

- 4) **Nalezněte různé hodnoty vyskytující se ve sloupci *Order.Order_Status* a k nim množství objednávek s daným statusem. Záznamy seřadte dle sloupce *Order_Status*.**

```
SELECT Order_Status, COUNT(*) AS Četnost
FROM [csv].[Order]
GROUP BY Order_Status
ORDER BY Order_Status
```

	Order_Status	Četnost
1	1	4666
2	2	1079
3	3	268

Záznamy tabulky seskupíme dle sloupce **Order_Status** pomocí klauzule **GROUP BY** a pomocí agregační funkce **COUNT** spočítáme **počet řádků s touto hodnotou**. Jelikož počítáme celé řádky, můžeme ve funkci použít **jakýkoliv sloupec** nebo i znak *****.

- 5) Na základě hodnot sloupce *Order.Order_Status* vytvořte číselníkovou tabulku *OrderStatus*, která bude obsahovat sloupce *OrderStatus_ID* (se stejným datovým typem jako *Order.Order_Status*) a *OrderStatus_Name* (textový řetězec s diakritikou a maximálně 20ti znaky). Po vytvoření tabulky do ní vložte záznamy ('1', 'Zaplacená'), ('2', 'Nezaplacená'), ('3', 'Zrušená').

```
CREATE TABLE OrderStatus (OrderStatus_ID CHAR(1), OrderStatus_Name NVARCHAR(20))
;
INSERT INTO OrderStatus
VALUES ('1', 'Zaplacená'), ('2', 'Nezaplacená'), ('3', 'Zrušená')
```

	OrderStatus_ID	OrderStatus_Name
1	1	Zaplacená
2	2	Nezaplacená
3	3	Zrušená

- 6) V tabulce *Order* zjistěte, jaká různá množství položek (sloupec *Quantity*) byla zatím zákazníci v rámci jedné objednávky objednána. Seřadte tyto záznamy od nejvyššího množství po to nejnižší.

```
SELECT DISTINCT Quantity
FROM [Database_1].[csv].[Order]
ORDER BY Quantity DESC
```

- 7) Nalezněte seznam TOP 10 zákazníků dle počtu provedených objednávek. Poté nalezněte také TOP 10 těch, kteří si objednali největší množství položek.

```
SELECT TOP(10) Customer_ID
, COUNT(*) as PočetObjednávek
FROM [csv].[Order]
GROUP BY Customer_ID
ORDER BY PočetObjednávek DESC
```

	Customer_ID	PočetObjednávek
1	C166	73
2	C203	72
3	C160	70
4	C124	69
5	C191	69
6	C183	69
7	C135	68
8	C168	68
9	C197	68
10	C151	68

Zatímto za účelem získání množství objednávek používáme funkci **COUNT** vracející ke každému zákazníkovi počet jeho záznamů, při sčítání položek bereme funkci **SUM**, ve které musíme uvést sloupec, jehož hodnoty sčítáme.

```
SELECT TOP(10) Customer_ID
, SUM(Quantity) as SoučetPoložek
FROM [csv].[Order]
GROUP BY Customer_ID
ORDER BY SoučetPoložek DESC
```

	Customer_ID	SoučetPoložek
1	C104	651
2	C204	567
3	C200	514
4	C154	463
5	C165	435
6	C180	404
7	C190	395
8	C169	385
9	C177	352
10	C120	351

Na základě této jednoduché analýzy můžeme například zjistit, že neexistuje ani jeden zákazník, který by patřil do obou skupin.

- 8) Záznamy v tabulce Order seskupte podle množství objednaných položek tak, abyste zjistili, jaká množství jsou objednávana nejčastěji.

```
SELECT Quantity AS Objednané_Množství
      ,COUNT(*) AS Četnost
FROM [csv].[Order]
GROUP BY Quantity
ORDER BY Četnost DESC
```

	Objednané_Množství	Četnost
1	2	2017
2	3	2005
3	1	1938
4	12	10
5	55	5

Z výsledku daného dotazu můžeme vyčíst, že objednávky v naprosté většině případů nejčastěji obsahují 2, 3 nebo jen 1 položku.

- 9) Záznamy v tabulce Order seskupte podle zákazníků a pro každého tak zjistěte:

- počet provedených objednávek
- celkovou sumu nakoupených položek
- počet **různých** produktů, které si zákazník kdy objednal
- datum první objednávky
- datum poslední objednávky

Zároveň záznamy seřadte **sestupně** podle **počtu provedených objednávek**.

```
SELECT Customer_ID
      ,COUNT(*) AS PočetObjednávek
      ,SUM(Quantity) AS SumaPoložek
      ,COUNT(DISTINCT Product_ID) AS PočetRůznýchProduktů
      ,MIN(Order_Date) AS PrvníObjednávka
      ,MAX(Order_Date) AS PosledníObjednávka
FROM [csv].[Order]
GROUP BY Customer_ID
ORDER BY PočetObjednávek DESC
```

	Customer_ID	PočetObjednávek	SumaPoložek	PočetRůznýchProduktů	PrvníObjednávka	PosledníObjednávka
1	C166	73	147	21	2020-02-28	2021-03-28
2	C203	72	137	19	2020-03-08	2021-03-27
3	C160	70	226	20	2020-02-27	2021-03-28
4	C124	69	140	21	2020-03-01	2021-03-28
5	C191	69	203	21	2020-02-25	2021-03-24
6	C183	69	145	21	2020-02-23	2021-03-27
7	C168	68	138	20	2020-02-23	2021-03-02

Pro získání počtu **různých** produktů vyskytujících se v objednávkách daného zákazníka je potřeba vložit klauzuli **DISTINCT** do funkce **COUNT**. Kdybychom **DISTINCT** nepoužili, byl by napočítán celkový počet hodnot **Product_ID**, který by se však v takovém případě rovnal počtu záznamů, a vyšel by tedy stejně jako **COUNT(*)**.

- 10) V tabulce Order naleznete kombinace zákazníka a konkrétního data, v rámci kterých byly provedeny více než 2 objednávky. Jinými slovy zjistěte, kteří zákazníci a ve kterých dnech provedli více než 2 objednávky. Získané záznamy zároveň **seřadte** tak, aby **nejnovější objednávky** byly v seznamu jako první.

```
SELECT Customer_ID
       ,Order_Date
       ,COUNT(Order_ID) AS PočetObjednávek
FROM [csv].[Order]
GROUP BY Customer_ID, Order_Date
HAVING COUNT(Order_ID) > 2
ORDER BY Order_Date DESC
```

	Customer_ID	Order_Date	PočetObjednávek
1	C176	2021-03-26	3
2	C187	2021-03-23	3
3	C125	2020-12-10	3
4	C127	2020-09-14	3
5	C139	2020-09-02	3
6	C118	2020-06-27	3
7	C204	2020-06-19	3
8	C106	2020-06-10	3
9	C131	2020-06-06	3
10	C190	2020-05-30	3
11	C186	2020-04-16	3
12	C171	2020-04-15	3

Pro seskupování záznamů dle **kombinace dvou sloupců** je potřeba oba zahrnout v klauzuli **GROUP BY**. Abychom poté mohli filtrovat dle hodnot agregovaných funkcí **COUNT**, musíme pravidlo zahrnout do klauzule **HAVING**.

- 11) Napište dotaz, který z tabulky Product vrátí všechny záznamy pro sloupce *Product_Name*, *Product_Category* a *Weight*, a navíc k nim přidá informaci o nejvyšší váze v rámci příslušné kategorie.

```
SELECT Product_Name
       ,Product_Category
       ,Weight
       ,MAX(Weight) OVER(PARTITION BY Product_Category) AS MAXVáhaKategorie
FROM csv.Product
```

Ve výsledku dotazu můžeme vidět, že každému produktu náležícímu do stejné kategorie je přiřazena stejná hodnota sloupce **MAXVáhaKategorie**, která odpovídá nejvyšší váze produktu v rámci této kategorie.

	Product_Name	Product_Category	Weight	MAXVáhaKategorie
1	Jeans	Clothes	562.00	672.00
2	Shirt	Clothes	315.00	672.00
3	Shorts	Clothes	672.00	672.00
4	T-Shirt	Clothes	243.00	672.00
5	Sofa	Furniture	3432.00	3432.00
6	Cupboard	Furniture	1932.00	3432.00

Obdobným způsobem bychom mohli spočítat také např. průměrnou váhu produktu v rámci dané kategorie.

```
SELECT Product_Name
       ,Product_Category
       ,Weight
       ,MAX(Weight) OVER(PARTITION BY Product_Category) AS MAXVáhaKategorie
       ,AVG(Weight) OVER(PARTITION BY Product_Category) AS AVGVáhaKategorie
FROM csv.Product
```

- 12) Na základě dotazu z minulé úlohy vypočítejte pro každý záznam tabulky **Product** rozdíl mezi *maximální váhou z dané produkové kategorie* a *váhou určitého produktu* na daném řádku.

Za účelem výpočtu rozdílu mezi **maximální váhou v rámci kategorie** a **váhou konkrétního produktu** stačí použít hodnotu sloupce **MAXVáhaKategorie** (z předchozí úlohy) a odečíst od ní hodnotu sloupce **Weight**, která na každém řádku odpovídá váze daného produktu.

```
SELECT Product_Name
,Product_Category
,Weight
,MAX(Weight) OVER(PARTITION BY Product_Category) AS MAXVáhaKategorie
,MAX(Weight) OVER(PARTITION BY Product_Category) - Weight AS VáhovýRozdíl
FROM csv.Product
```

	Product_Name	Product_Category	Weight	MAXVáhaKategorie	VáhovýRozdíl
1	Jeans	Clothes	562.00	672.00	110.00
2	Shirt	Clothes	315.00	672.00	357.00
3	Shorts	Clothes	672.00	672.00	0.00
4	T-Shirt	Clothes	243.00	672.00	429.00
5	Sofa	Furniture	3432.00	3432.00	0.00
6	Cupboard	Furniture	1932.00	3432.00	1500.00
7	Blanket	Homeware	744.00	1570.00	826.00
8	Picture	Homeware	1570.00	1570.00	0.00
9	Decoration	Homeware	679.00	1570.00	891.00
10	Carpet	Homeware	554.00	1570.00	1016.00

- 13) Napište dotaz, který vrátí všechny záznamy z tabulky **Order**, přičemž ke každému zákazníkovi přiřadí informaci o jeho celkovém počtu objednávek. Data navíc **seřadíte** podle **kódu zákazníka**.

```
SELECT *
, COUNT(Order_ID) OVER(PARTITION BY Customer_ID) AS PočetObjednávek
FROM [csv].[Order]
ORDER BY Customer_ID
```

Ke každému řádku je přiřazen příslušný **počet objednávek** daného **zákazníka**. Ve výsledku dotazu potom můžeme vidět, že např. u zákazníka C100, kterému byla přiřazena hodnota 64, opravdu v tabulce napočítáme 64 záznamů (objednávek).

60	O6725	2021-03-13	C100	P103	3	1	CZK	64
61	O6780	2020-12-17	C100	P114	2	2	CZK	64
62	O6805	2020-07-20	C100	P104	1	1	CZK	64
63	O6919	2020-11-23	C100	P120	3	3	CZK	64
64	O6928	2020-10-20	C100	P115	3	3	CZK	64
65	O6825	2020-06-16	C101	P107	1	2	CZK	63
66	O6839	2020-11-03	C101	P111	1	2	CZK	63
67	O6806	2021-02-23	C101	P107	3	3	CZK	63
68	O6711	2020-10-25	C101	P105	2	1	CZK	63
69	O6697	2021-01-25	C101	P102	2	2	CZK	63