

Cvičení 3: DDL – CREATE, ALTER, TRUNCATE a DROP

- 1) Do tabulky **Product** přidejte sloupce **Length**, **Width** a **Height** odpovídající délce, šířce a výšce produktu v centimetrech. Všechny by měly být čísla maximálně v řádu tisíců a se dvěma desetinnými místy.
- 2) Vytvořte novou tabulku **csv.CustomerType** se sloupci **CustomerType_ID** (primární klíč), **CustomerType_Name** a **CustomerType_Description**. První sloupec bude obsahovat vždy hodnoty v podobě jednoho písmene. Zbývající dva sloupce mohou být plněny textovými řetězci s diakritikou až do délky 100 znaků. Tato tabulka bude později sloužit jako číselník zákaznických kategorií.
- 3) Do tabulky **Customer** přidejte sloupec **CustomerType_ID**. Ten by měl svými hodnotami odpovídat stejnojmennému sloupci z nově vytvořené tabulky **CustomerType**.
- 4) V tabulce **Customer** změňte datový typ sloupce pro datum registrace tak, aby se v něm odted ukládaly hodnoty ve formátu data i času. Dále v této tabulce upravte i datový typ sloupce **Address**. Podle aktuálních dat od klienta jste totiž zjistili, že adresy mohou být až 250 znaků dlouhé.
- 5) Do nově vytvořené tabulky **CustomerType** přidejte omezení, že sloupec **CustomerType_Name** nesmí obsahovat prázdné hodnoty.
- 6) Pro sloupec **Contact.Email** (tedy sloupec **Email** z tabulky **Contact**) vytvořte integritní omezení s názvem **DF_ContactEmail**. To bude určovat výchozí hodnotu pro tento sloupec v podobě slova 'Unknown'.
- 7) Pro sloupce **Contact.Contact_Name**, **Contact.Contact_Surname** a **Contact.Phone_Number** vytvořte unikátní index (unikátní klíč) s názvem **UK_NameSurnamePhone**.

- 8) Hodnoty sloupce *Supplier.Supplier_Contact* by měly vždy odpovídat některé z hodnot sloupce *Contact.Contact_ID* v číselníku kontaktů. Vytvořte tedy pro tuto vazbu cizí klíč s názvem ***FK_Supplier_Contact***.
- 9) Do nově vytvořené tabulky *CustomerType* vložte manuálně cvičné záznamy s hodnotami (1, 'CV1', 'Cvičný záznam 1') a (2 'CV2', 'Cvičný záznam 2'). Využijte k tomu možnost ***Edit Top 200 Rows*** po kliknutí pravým tlačítkem myši na název dané tabulky.
- 10) Nyní se pomocí možnosti ***Select Top 1000 Rows*** ujistěte, že vytvořené záznamy opravdu v tabulce *CustomerType* existují. Poté využijte výraz **TRUNCATE** k vymazání dat z této tabulky.
- 11) Rozhodli jsme se, že nepotřebujeme sloupec *Customer.Phone_Prefix*, proto jej pomocí výrazu **DROP** můžete smazat. Stejně tak je možné odstranit i integritní omezení definující výchozí hodnotu pro sloupec *Customer.City*.

Cvičení 3: DDL – CREATE, ALTER, TRUNCATE a DROP

- 1) Do tabulky **Product** přidejte sloupce **Length**, **Width** a **Height** odpovídající délce, šířce a výšce produktu v centimetrech. Všechny by měly být čísla maximálně v řádu tisíců a se dvěma desetinnými místy.

Řešením je následující dotaz. Můžete si všimnout, že slovo **Length** v něm svítí modře, protože ho SQL rozpoznalo jako stejnojmennou funkci. Když jsem za tímto výrazem ale nepoužili závorky s určitými argumenty funkce, a místo toho jsme definovali datový typ, SQL jej správně pochopilo jako název nového sloupce, ale zabarvení v okně dotazu zůstalo.

```
ALTER TABLE [csv].[Product]
ADD Length DECIMAL(6,2), Width DECIMAL(6,2), Height DECIMAL(6,2)
```

- 2) Vytvořte novou tabulku **csv.CustomerType** se sloupci **CustomerType_ID** (primární klíč), **CustomerType_Name** a **CustomerType_Description**. První sloupec bude obsahovat vždy hodnoty v podobě jednoho písmene. Zbývající dva sloupce mohou být plněny textovými řetězci s diakritikou až do délky 100 znaků. Tato tabulka bude později sloužit jako číselník zákaznických kategorií.

Řešením je následující dotaz. Aby byla po úspěšném spuštění dotazu tabulka vidět v databázi, je potřeba vždy zvolit možnost **Refresh** po kliknutí pravým tlačítkem na záložku **Tables**. Pokud provádíme změny na již existující tabulce, stačí tímto způsobem aktualizovat pouze ji (kliknutím pravým tlačítkem na její název).

```
CREATE TABLE csv.CustomerType (CustomerType_ID CHAR(1) PRIMARY KEY
                                ,CustomerType_Name NVARCHAR(100)
                                ,CustomerType_Description NVARCHAR(100)
                                )
```

- 3) Do tabulky **Customer** přidejte sloupec **CustomerType_ID**. Ten by měl svými hodnotami odpovídat stejnojmennému sloupci z nově vytvořené tabulky **CustomerType**.

```
ALTER TABLE [csv].Customer  
ADD CustomerType_ID CHAR(1)
```

- 4) V tabulce **Customer** změňte datový typ sloupce pro datum registrace tak, aby se v něm odteď ukládaly hodnoty ve formátu data i času. Dále v této tabulce upravte i datový typ sloupce **Address**. Podle aktuálních dat od klienta jste totiž zjistili, že adresy mohou být až 250 znaků dlouhé.

Řešením je následující dotaz, který je složený ze dvou částí oddělených středníkem. Pokud celý takový dotaz spustíme naráz, SQL provede nejdříve ten první, a potom ten druhý. Nemusíme tedy spouštět každý zvlášť.

```
ALTER TABLE [csv].Customer  
ALTER COLUMN Date_Registered DATETIME  
;  
ALTER TABLE [csv].Customer  
ALTER COLUMN Address NVARCHAR(250)
```

- 5) Do nově vytvořené tabulky **CustomerType** přidejte omezení, že sloupec **CustomerType_Name** nesmí obsahovat prázdné hodnoty.

```
ALTER TABLE [csv].CustomerType  
ALTER COLUMN CustomerType_Name NVARCHAR(100) NOT NULL
```

- 6) Pro sloupec **Contact.Email** (tedy sloupec **Email** z tabulky **Contact**) vytvořte integritní omezení s názvem **DF_ContactEmail**. To bude určovat výchozí hodnotu pro tento sloupec v podobě slova 'Unknown'.

```
ALTER TABLE [csv].Contact  
ADD CONSTRAINT DF_ContactEmail DEFAULT 'Unknown' FOR Email
```

- 7) Pro sloupce *Contact.Contact_Name*, *Contact.Contact_Surname* a *Contact.Phone_Number* vytvořte unikátní index (unikátní klíč) s názvem *UK_NameSurnamePhone*.

Řešením je následující dotaz vytvářející unikátní index nad sloupci se jménem, příjmením a telefonním číslem kontaktů v tabulce **Contact**. Tím je zaručeno, že do tabulky nebude možné vložit duplicitní záznamy, které by měly hodnoty v těchto sloupcích shodné s již existujícím řádkem v tabulce.

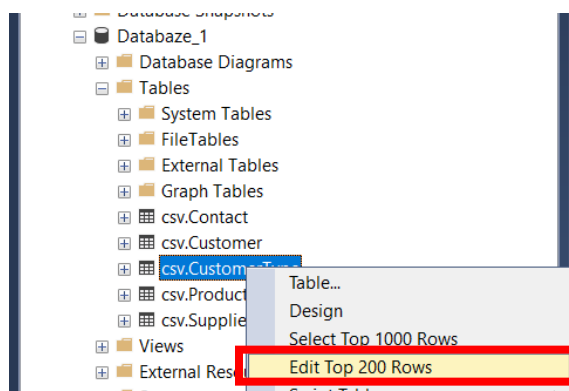
```
ALTER TABLE [csv].Contact
ADD CONSTRAINT UK_NameSurnamePhone
UNIQUE (Contact_Name, Contact_Surname, Phone_Number)
```

- 8) Hodnoty sloupce *Supplier.Supplier_Contact* by měly vždy odpovídat některé z hodnot sloupce *Contact.Contact_ID* v číselníku kontaktů. Vytvořte tedy pro tuto vazbu cizí klíč s názvem *FK_Supplier_Contact*.

```
ALTER TABLE csv.Supplier
ADD CONSTRAINT FK_Supplier_Contact
FOREIGN KEY (Supplier_Contact) REFERENCES csv.Contact(Contact_ID)
```

- 9) Do nově vytvořené tabulky *CustomerType* vložte manuálně cvičné záznamy s hodnotami (1, 'CV1', 'Cvičný záznam 1') a (2 'CV2', 'Cvičný záznam 2'). Využijte k tomu možnost **Edit Top 200 Rows** po kliknutí pravým tlačítkem myši na název dané tabulky.

Po zvolení možnosti **Edit Top 200 Rows** napíšeme příslušné hodnoty do příslušných sloupců zobrazené tabulky. Nejprve se u nich zobrazí vykřičníky, ale když celý záznam dopíšeme a klikneme napředchozí či další řádek, vykřičníky zmizí a daný záznam se **automaticky uloží**.

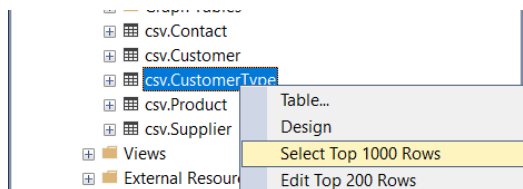


	CustomerT...	CustomerT...	CustomerType_De...
	1	CV1	Cvičný záznam 1
	2	CV2	Cvičný záznam 2
*	NULL	NULL	NULL

DESKTOP-KN4RSRP....csv.CustomerType			SQLQuery
	CustomerT...	CustomerT...	CustomerType_De...
	1	CV1	Cvičný záznam 1
	2	CV2	Cvičný záznam 2
▶*	NULL	NULL	NULL

- 10) Nyní se pomocí možnosti **Select Top 1000 Rows** ujistěte, že vytvořené záznamy opravdu v tabulce **CustomerType** existují. Poté využijte výraz **TRUNCATE** k vymazání dat z této tabulky.

Po zvolení možnosti **Select Top 1000 Rows** vidíme, že vytvořené řádky v tabulce **CustomerType** opravdu byly uloženy.

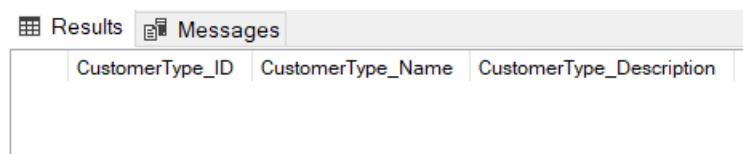


	CustomerType_ID	CustomerType_Name	CustomerType_Description
1	1	CV1	Cvičný záznam 1
2	2	CV2	Cvičný záznam 2

Nyní můžeme použít výraz **TRUNCATE** pro opětovné vymazání všech řádků tabulky.

```
TRUNCATE TABLE [csv].[CustomerType]
```

Po znovu spuštění **Select Top 1000 Rows** vidíme, že je tabulka prázdná.



CustomerType_ID	CustomerType_Name	CustomerType_Description
-----------------	-------------------	--------------------------

- 11) Rozhodli jsme se, že nepotřebujeme sloupec **Customer.Phone_Prefix**, proto jej pomocí výrazu **DROP** můžete smazat. Stejně tak je možné odstranit i integritní omezení definující výchozí hodnotu pro sloupec **Customer.City**.

Pokud neznáme název integritního omezení zmíněného v zadání, můžeme se podívat do záložky **Constraints**.

```
ALTER TABLE csv.Customer
DROP COLUMN Phone_Prefix
;
ALTER TABLE csv.Customer
DROP CONSTRAINT DF_Customer_City
```

