

## Cvičení 1: DDL – CREATE

- 1) Vytvořte v SSMS databázi libovolného názvu a v ní poté schéma „csv“.
- 2) Vytvořte v dané databázi a daném schématu následující tabulky.

## a) Supplier

Tabulka bude obsahovat data o dodavatelích firmy v rámci atributů:

- **kód dodavatele** (Supplier\_Code)
- **jméno dodavatele** (Supplier\_Name)
- **kód země** (Country\_Code)
- **kód kontaktní osoby** (Supplier\_Contact)
- **datum začátku spolupráce s dodavatelem** (Date\_Start)
- **datum ukončení spolupráce s dodavatelem** (Date\_End)
- **počet různých produktů odebíraných od dodavatele** (Product\_Amount)
- **cena nejlevnějšího dodávaného produktu** (Lowest\_Price)
- **cena nejdražšího dodávaného produktu** (Highest\_Price)
- **měna dodavatele** (Supplier\_Currency)

Atributy **kód dodavatele** (**primární klíč** tabulky) a **kód kontaktní osoby** budou obsahovat textové řetězce o fixní délce 6 znaků, zatímco **kód země** a **měna dodavatele** budou vždy třípísmenné. **Jméno dodavatele** může obsahovat diakritiku a má povolenou libovolnou délku, ne však větší než 150 znaků. Všechny tyto sloupce budou v tabulce vždy plněny **neprázdnou hodnotou**.

Do **datumových sloupců** budeme ukládat pouze informace o datu, tedy bez času. Jejich **výchozí hodnotu** nastavíme na 1.1.1900, abychom ve sloupcích neměli prázdné hodnoty NULL, ani když příslušné datum neznáme.

Horní hranici **počtu různých produktů odebíraných od dodavatele** v tuto chvíli neznáme, ale pravděpodobně bude v řádu stovek. Nejnížší a nejvyšší **ceny dodávaných produktů** chceme ukládat s jedním desetinným místem (v žádné měně by zde neměla být hodnota vyšší než 30 000).

## b) **Contact**

Tabulka bude obsahovat kontakty firmy v rámci atributů:

- **kód kontaktu** (Contact\_ID)
- **jméno osoby** (Contact\_Name)
- **příjmení osoby** (Contact\_Surname)
- **kód země** (Country\_Code)
- **telefonní předvolba** (Phone\_Prefix)
- **telefonní číslo** (Phone\_Number)
- **email** (Email)

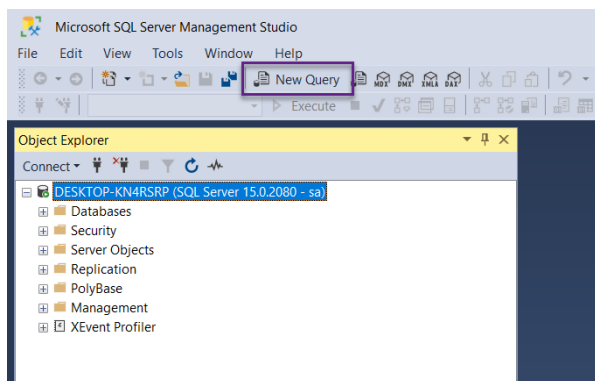
Atribut **kód kontaktu**, který je **primárním klíčem** tabulky, bude obsahovat textové řetězce o fixní délce 6 znaků, zatímco **kód země** a **telefonní předvolba** budou vždy složené ze 3 znaků. **Jméno, příjmení i email kontaktu** budou řetězce s diakritikou či různými speciálními znaky a libovolnou délkou, ne však delší než 150 znaků. **Telefonní číslo** by mělo být **omezeno** tak, aby obsahovalo buď 9 nebo 10 cifer (délka čísla se může lišit dle země kontaktu).

Všechny sloupce tabulky kromě **emailu** budou muset být vždy vyplněny **neprázdnou hodnotou**.

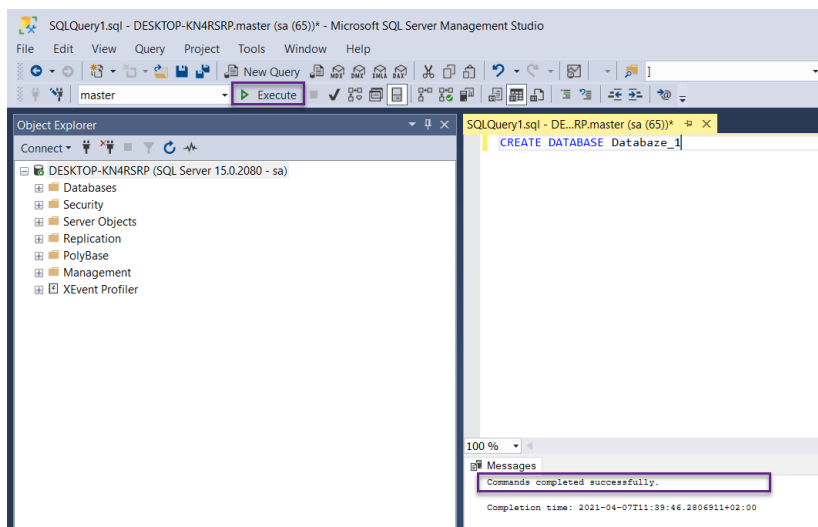
## Cvičení 1: DDL – CREATE

1) Vytvořte v SSMS databázi libovolného názvu a v ní poté schéma „csv“.

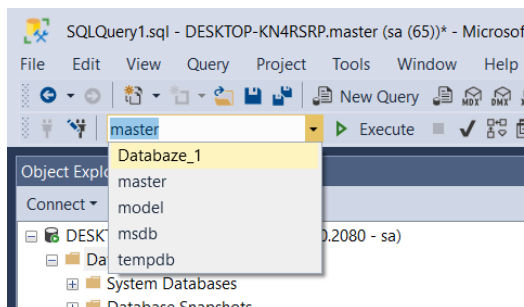
Po přihlášení do SSMS si otevřete nové okno dotazu pomocí tlačítka **New Query**. Stejnou operaci lze provést také klávesovou zkratkou **CTRL + N**.



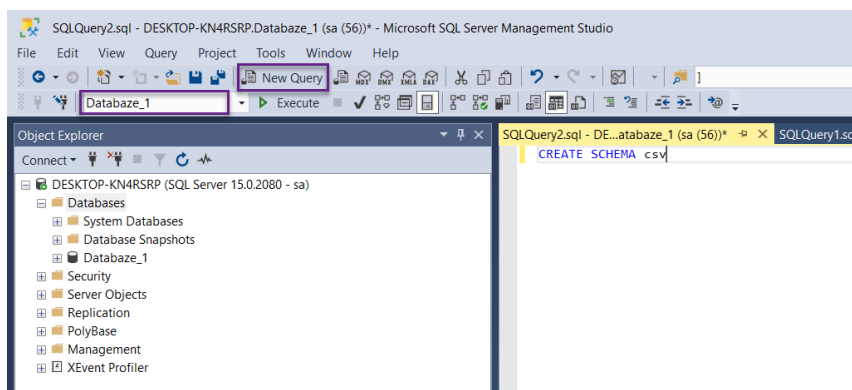
Poté do okna můžete napsat jakýkoliv dotaz – v našem případě **CREATE DATABASE** s názvem vytvářené databáze. Spustit jej lze tlačítkem **Execute** nebo klávesou **F5**. O úspěšném provedení dotazu nás SSMS informuje zprávou **Commands completed successfully**.



Za účelem vytvoření **schématu** či jakéhokoliv **objektu dané databáze** je v dalším kroku potřeba tuto databázi zvolit v **rozbalovacím seznamu** vedle tlačítka **Execute**. Po každém spuštění SSMS je databáze v tomto okně automaticky přepnuta zpět na **master**.



Nyní lze v novém okně vytvářet databázová schémata (v našem případě schéma s názvem **csv**), databázové objekty či spouštět jiné libovolné dotazy.



## 2) Vytvořte v dané databázi a daném schématu následující tabulky.

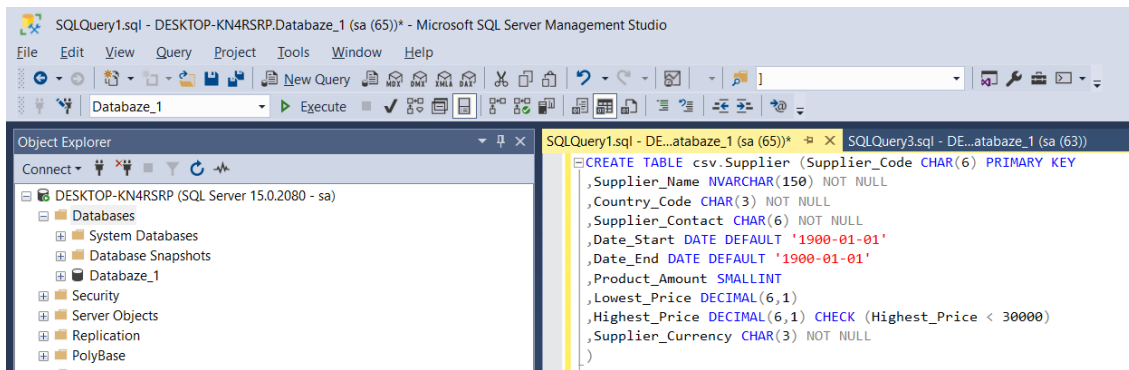
### a) **Supplier**

Tabulku dodavatelů vytvoříme pomocí příkazu **CREATE TABLE**, kde před samotným **názvem tabulky** napíšeme také **schéma**, ve kterém bude tabulka uložena, tedy **csv.Supplier**. Kdybychom název schématu nespecifikovali, bude tabulka uložena do toho výchozího, tedy **dbo**.

Dále v dotazu vypíšeme **jednotlivé sloupce** s příslušnými **datovými typy** dle požadavků na povolené hodnoty definované v zadání. Poté také doplníme konkrétní **integritní omezení**:

- **primární klíč** pro sloupec **Supplier\_Code**
- **NOT NULL** pro sloupce **Supplier\_Name**, **Country\_Code**, **Supplier\_Contact** a **Supplier\_Currency** (sloupec **Supplier\_Code** již takto omezen je díky primárnímu klíči)
- **DEFAULT** pro sloupce **Date\_Start** a **Date\_End**
- **CHECK** pro sloupec **Highest\_Price**

Výsledný dotaz poté může vypadat následovně.



Pokud bychom dotaz chtěli (jen pro přehlednost) upravit tak, aby byly všechny sloupce vypsané pod **Supplier\_Code**, je možné to udělat následovně:

- Klikneme levým tlačítkem myši na začátek druhého řádku, kam tím přesuneme blikající kurzor.
- Stiskneme a držíme klávesy **SHIFT + ALT** a zároveň devětkrát stiskneme také šipku dolů, čímž označíme všechny zbývající řádky dotazu.
- Poté klávesy pustíme a pomocí **TAB** či **mezerníku** posouváme všechny označené řádky směrem doprava. Tímto způsobem lze před všechny řádky také např. napsat nějaký znak nebo s nimi provádět jiné operace.

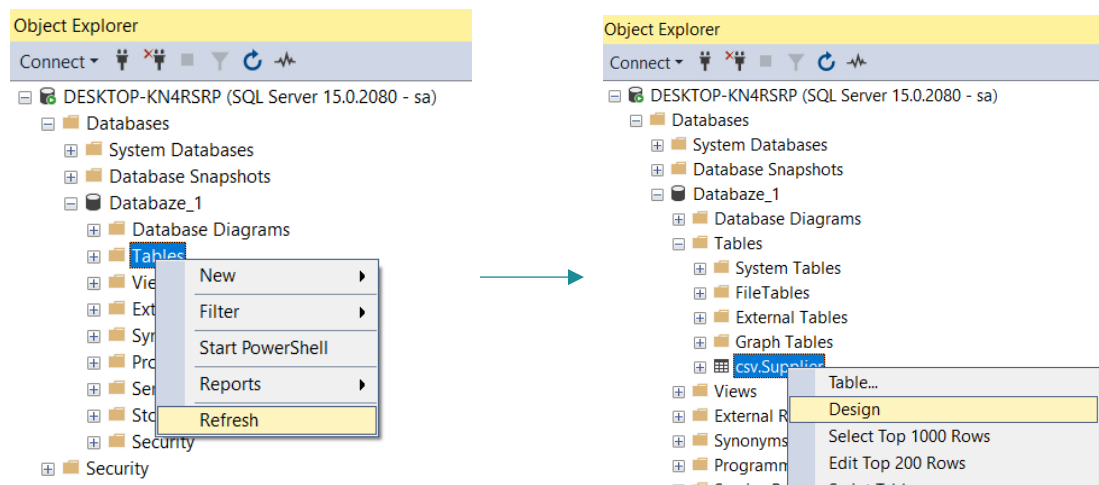
Výsledný dotaz po zarovnání řádků vypadá následovně.

```
CREATE TABLE csv.Supplier (Supplier_Code CHAR(6) PRIMARY KEY
,Supplier_Name NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Supplier_Contact CHAR(6) NOT NULL
,Date_Start DATE DEFAULT '1900-01-01'
,Date_End DATE DEFAULT '1900-01-01'
,Product_Amount SMALLINT
,Lowest_Price DECIMAL(6,1)
,Highest_Price DECIMAL(6,1) CHECK (Highest_Price < 30000)
,Supplier_Currency CHAR(3) NOT NULL
)

CREATE TABLE csv.Supplier (Supplier_Code CHAR(6) PRIMARY KEY
,Supplier_Name NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Supplier_Contact CHAR(6) NOT NULL
,Date_Start DATE DEFAULT '1900-01-01'
,Date_End DATE DEFAULT '1900-01-01'
,Product_Amount SMALLINT
,Lowest_Price DECIMAL(6,1)
,Highest_Price DECIMAL(6,1) CHECK (Highest_Price < 30000)
,Supplier_Currency CHAR(3) NOT NULL
)
```

Funkce dotazu se touto estetickou úpravou nijak nezmění, jen je dobré mít na paměti, že SQL skripty, které v databázích vytváříme, po nás často čtou další uživatelé. Proto by námi vytvářené dotazy měly být snadno čitelné a napsané ve standardní podobě dle „best practice“ pravidel, které má daná společnost definována.

Po samotném spuštění dotazu a jeho úspěšném zpracování stačí už jen zvolit možnost **Refresh** v **Object Explorer** záložce **Databases – Tables**, a vytvořená tabulka je k dispozici pro další práci a úpravy. Pokud bychom chtěli měnit její strukturu (názvy sloupců, datové typy, primární klíče, apod.), můžeme to udělat pomocí možnosti **Design** po kliknutí na název tabulky pravým tlačítkem myši.



## b) Contact

Tabulku firemních kontaktů vytvoříme obdobně jako tabulku dodavatelů pomocí příkazu **CREATE TABLE**, definovaného **schématu** a odpovídajícího **názvu** objektu, tedy **csv.Contact**.

Dále v dotazu vypíšeme **jednotlivé sloupce** s příslušnými **datovými typy** (dle požadavků na povolené hodnoty definované v zadání) spolu s konkrétními **integritními omezení**:

- **primární klíč** pro sloupec **Contact\_ID**
- **NOT NULL** pro sloupce **Contact\_Name**, **Contact\_Surname**, **Country\_Code**, **Phone\_Prefix** a **Phone\_Number** (sloupec **Contact\_ID** již takto omezen je díky primárnímu klíči)

Výsledný dotaz pro vytvoření této tabulky tedy může vypadat následovně.

```
CREATE TABLE csv.Contact (Contact_ID CHAR(6) PRIMARY KEY
,Contact_Name NVARCHAR(150) NOT NULL
,Contact_Surname NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Phone_Prefix CHAR(3) NOT NULL
,Phone_Number BIGINT NOT NULL CHECK (Phone_Number >= 1000000 AND Phone_Number <= 9999999999)
,Email NVARCHAR(150)
)
```

Všimněte si, že pro sloupec **Phone\_Number** byl zvolen datový typ **BIGINT**, jelikož žádný menší číselný formát není schopný vracet **desetimístná celá čísla** větší než 2 147 483 648 (horní hranice datového typu **INT**).

Zároveň v dotazu definujeme **CHECK** podmínku omezující hodnoty tohoto sloupce tak, aby byly mezi **nejmenším devítimístným** a **největším desetimístným číslem**.

To lze udělat také pomocí operátoru **BETWEEN**, který si blíže představíme v dalším lekcích.

```
CREATE TABLE csv.Contact (Contact_ID CHAR(6) PRIMARY KEY
,Contact_Name NVARCHAR(150) NOT NULL
,Contact_Surname NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Phone_Prefix CHAR(3) NOT NULL
,Phone_Number BIGINT NOT NULL CHECK (Phone_Number BETWEEN 1000000 AND 9999999999)
,Email NVARCHAR(150)
)
```

Alternativní řešení liší se v **datovém typu** a **CHECK** podmínkách pro sloupec **Phone\_Number** je následující.

```
CREATE TABLE csv.Contact (Contact_ID CHAR(6) PRIMARY KEY
,Contact_Name NVARCHAR(150) NOT NULL
,Contact_Surname NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Phone_Prefix CHAR(3) NOT NULL
,Phone_Number VARCHAR(10) NOT NULL CHECK (LEN(Phone_Number) > 8)
,Email NVARCHAR(150)
)
```

Datový typ pro sloupec **Phone\_Number** je definován jako **VARCHAR(10)**, což znamená, že toto pole bude moci být plněno maximálně deseti znaky (čísly, písmeny, či jakýmkoliv jiným znakem).

Ke sloupci navíc přidáváme **CHECK** omezení s funkcí **LEN**, která vrací délku daného řetězce. Tou stanovíme, že délka řetězců ve sloupci **Phone\_Number** musí být větší než 8. Spolu s maximální délkou řetězce definovanou datovým typem **VARCHAR(10)** tedy hodnoty omezujeme na 9 až 10 znaků.

Kdybychom však chtěli mít jistotu, že se sloupec plní čistě číselnými hodnotami, a ne jinými znaky, můžeme přidat dodatečnou podmínku s funkcí **ISNUMERIC**, která vrací hodnotu 1, pokud je daný výraz číslem, nebo 0, pokud číslem není.

Když tedy v rámci **CHECK** podmínky definujeme, že výsledek funkce **ISNUMERIC** z hodnot sloupce **Phone\_Number** musí být roven 1, potom jakákoliv nečíselná hodnota vkládaná do tohoto sloupce bude zamítnuta.

```
CREATE TABLE csv.Contact (Contact_ID CHAR(6) PRIMARY KEY
,Contact_Name NVARCHAR(150) NOT NULL
,Contact_Surname NVARCHAR(150) NOT NULL
,Country_Code CHAR(3) NOT NULL
,Phone_Prefix CHAR(3) NOT NULL
,Phone_Number VARCHAR(10) NOT NULL CHECK (LEN(Phone_Number) > 8 AND ISNUMERIC(Phone_Number) = 1)
,Email NVARCHAR(150)
)
```

Různé typy SQL funkcí včetně **LEN** a **ISNUMERIC** si blíže popíšeme v poslední části kurzu na příkladech komplexnějších dotazů.