

# 1. Filmová databáze III

Autor: Ondřej Kořínek

Oblast z RVP Doporučený stupeň	Informační systémy Střední škola
Naplněvané výstupy RVP	<ul style="list-style-type: none"><li>identifikuje jejich jednotlivé prvky a vztahy mezi nimi; zvažuje možná rizika při navrhování i užívání informačních systémů</li><li>nastavuje zobrazení, řazení a filtrování dat v tabulce, aby mohl odpovědět na položenou otázku; využívá funkce pro automatizaci zpracování dat</li><li>vymezí problém a určí, jak při jeho řešení využije evidenci dat; na základě doporučeného i vlastního návrhu sestaví tabulku pro evidenci dat a nastaví pravidla a postupy pro práci se záznamy v evidenci dat</li><li>sám evidenci vyzkouší a následně zhodnotí její funkčnost, případně navrhne její úpravu</li></ul>
Vstupní požadavky na žáky	Vycházející z předchozího příspěvku Filmová databáze II
Cíl aktivity	<ul style="list-style-type: none"><li>Žáci nainstalují a nastaví SQL Server a SQL Server Management Studio</li><li>Žáci naimportují databáze z programu MS ACCESS na SQL Server pomocí SQL Server Management Studia</li><li>Žáci vytvoří kardinality vztahu mezi jednotlivými tabulkami pomocí možnosti Design</li><li>Žáci vytvoří struktury tabulek pomocí možnosti Design</li><li>Žáci vytvoří tabulky a kardinality vztahu mezi jednotlivými tabulkami pomocí jazyku SQL</li></ul>
Rozvíjené kompetence	<p>Klíčové kompetence prolínající se s rozvojem digitální gramotnosti žáků:</p> <p><b>Kompetence k učení</b></p> <ul style="list-style-type: none"><li>Samostatně/ případně s pomocí vyučujícího / pomocí diskuze s žáky ve skupině zpracovává příslušné zadání</li><li>Hodnotí vlastní řešení s předloženým řešením a navrhuje další možné zlepšování řešení</li><li>Využívá různé zdroje informací (internet, vyučující, vlastní znalosti, diskuze se spolužáky)</li></ul> <p><b>Kompetence k řešení problémů</b></p> <ul style="list-style-type: none"><li>Navrhuje entity/tabulky</li><li>Navrhuje strukturu tabulek</li><li>Navrhuje správnou kardinalitu mezi tabulkami</li><li>Navrhuje úpravu chyb</li></ul> <p><b>Kompetence komunikativní</b></p> <ul style="list-style-type: none"><li>Vysvětlí navržený databázový systém</li></ul> <p><b>Kompetence sociální a personální</b></p> <ul style="list-style-type: none"><li>při vytvoření tabulky/tabulek / kardinalit vztahů / entit spolupracuje se spolužáky</li></ul>
Potřebné vybavení	<ul style="list-style-type: none"><li>PC, internet, webový prohlížeč, MS ACCESS; žáci musí mít roli administrátora stanice, aby mohli instalovat nový SW, možno vyřešit pomocí dočasných virtuálních strojů</li></ul>

	<ul style="list-style-type: none"> <li>• SQL Server Management Studio + SQL Server</li> </ul>
Časová dotace	4 vyučovací hodiny

## Průběh výuky

### Uvedení do problematiky

Tento příspěvek je doplňkem příspěvku Filmová databáze II a uvádí možnost, jak databázi importovat z programu MS ACCESS na SQL Server, případně na SQL Serveru přímo vytvořit dané tabulky a kardinalitu mezi nimi pomocí jazyku SQL (DDL). Vyučujícím může být zvolena jedna z daných možností, buď bod 2, nebo bod 3.

### Zadání pro žáky

1. Nainstalujte si SQL Server a SQL Server Management Studio.
2. Nainportujte filmovou databázi, kterou jste vytvořili v programu MS ACCESS na SQL Server pomocí SQL Server Management Studia. Nainportujte pouze tabulky, danou kardinalitu mezi tabulkami vytvořte pomocí možnosti Design u jednotlivých tabulek.
3. Filmovou databázi podle zadání z příspěvku Filmová databáze II vytvořte celou na SQL Serveru pomocí jazyku SQL.

## Příprava na výuku

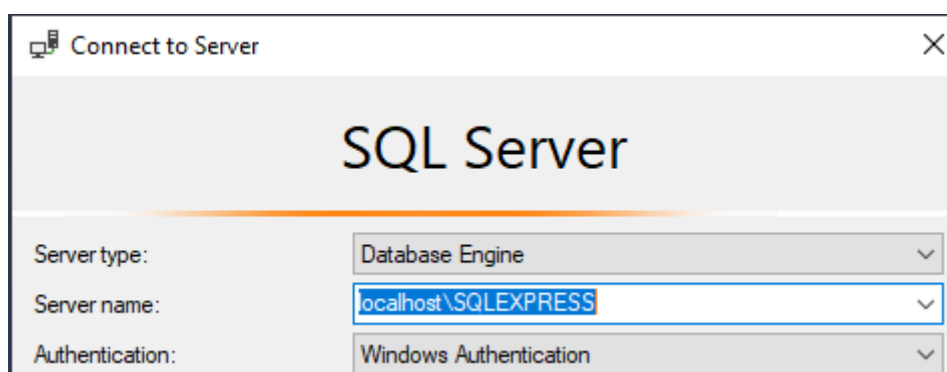
1. Instalace by neměla být nikterak složitá, stačí do vyhledávače zadat SQL Server a zvolit nainstalovat např. edici Express a Download SQL Server Management Studio (SSMS), stáhnout a nainstalovat. Při instalaci SQL Serveru je asi nejlepší vybrat typ instalace Basic a pokračovat v doporučeném nastavení. Po nainstalování se zobrazí dialogové okno, které je důležité. Žáci by si měli napsat jméno serveru, aby se k němu potom v počítači dokázali připojit, viz obrázek.



Obrázek 1 – Název serveru; Důležité je celý název vpravo za rovnítkem

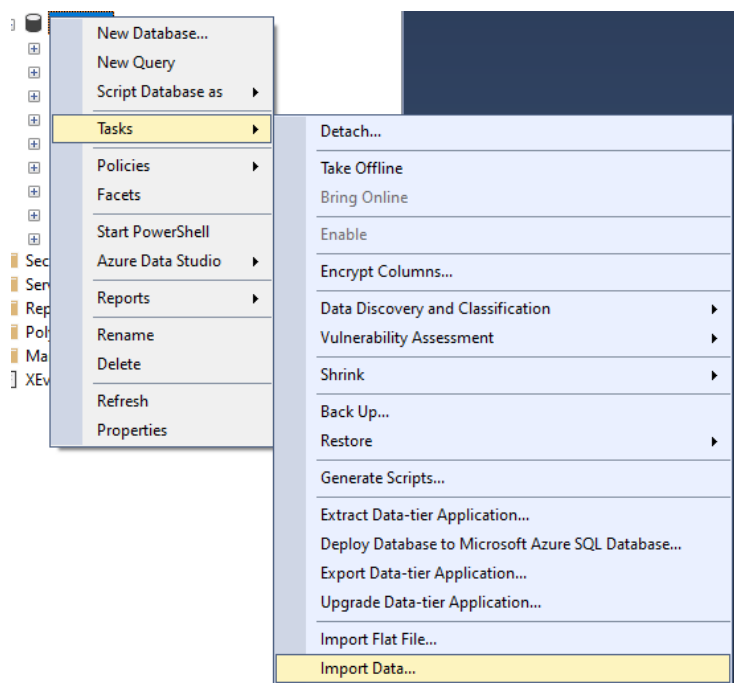
Navíc, níže v dialogovém okně je možnost nainstalovat SQL Server Management Studio (SSMS), které nemusejí žáci sami vyhledávat.

Při spuštění Management Studia se zadají/nastaví údaje uvedené na následujícím obrázku. Při lokálním použití na domácím PC doporučuji Windows autentizaci.



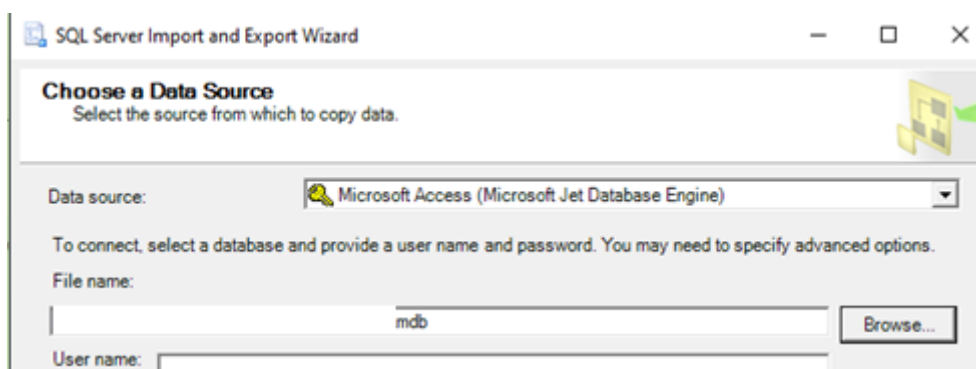
Obrázek 2 – Nastavení údajů pro přihlášení do Management Studia

2. Níže je uveden postup a upozornění na možné chyby importu, které se mohou přihodit. Je možné, že by si žáci mohli daný postup na internetu najít sami, případně je může vyučující nasměrovat. Je možné, že řešení nemusí být pouze níže uvedené. V postupu jsou uvedeny pouze základní kroky, pokud nějaký krok není uveden, stačí zvolit možnost Next.
  - a. Databázi, kterou máte v programu MS ACCESS, převedte na typ Databáze aplikace Access 2000. Výsledný soubor bude mít příponu: Microsoft Access Database (.mdb). Převod typu musí být proveden z programu, nelze příponu pouze mechanicky přepsat.
  - b. Připojte se na SQL Server pomocí programu Microsoft SQL Server Management Studio. V něm klikněte pravým tlačítkem myši na svoji (vybranou databázi), zvolte Tasks a Import data, uvedeno níže.



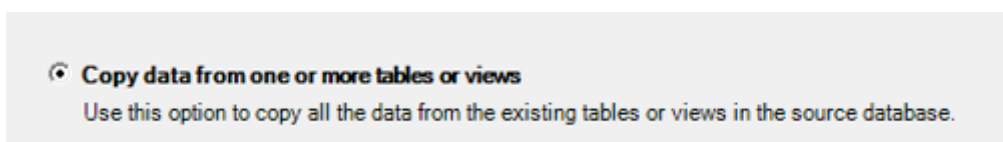
Obrázek 3 – Vybrání importu pomocí SQL Server Management Studia

- c. Vyberte Microsoft Access a pomocí Browse vyberte příslušnou databázi ve verzi 2000.



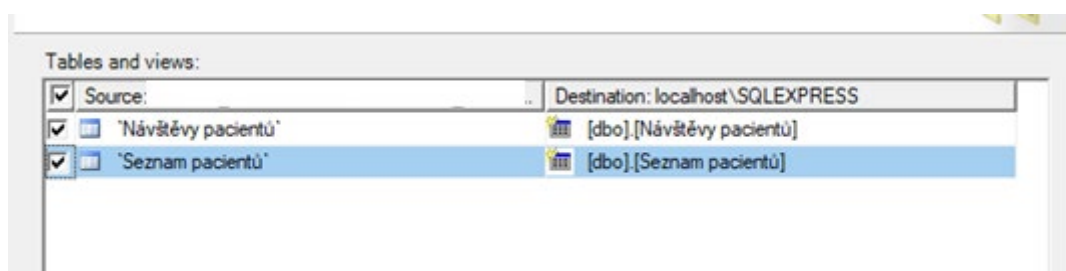
Obrázek 4 – Vybrání databáze programu MS ACCESS

- d. Vyberte možnost Microsoft OLE DB Provider for SQL Server. Mějte správný název serveru. Podle toho, jak se připojíte, vyberte příslušnou Authentication. Pokud máte SQL Server nainstalován na počítači lokálně, zvolte autentizaci Windows.
- e. Vyberte první možnost



Obrázek 5 – vybrání správné možnosti

- f. Zaškrtněte tabulky, které chcete importovat. Pokud je v nějaké tabulce v ACCESSU atribut typu Datum, je důležité v tomto kroku kliknout na ikonu tabulky a nastavit pro import místo datového typu Datetime datový typ Date. Pokud by se tak nestalo, je možné, že by se data do tabulky nenainportovala, vytvořila by se pouze její struktura. Pokud by import neproběhl, ale struktura tabulky by se na SQL Serveru vytvořila, musel by se import takové tabulky provést znovu, ale struktura tabulky na SQL Serveru by se musela odstranit.

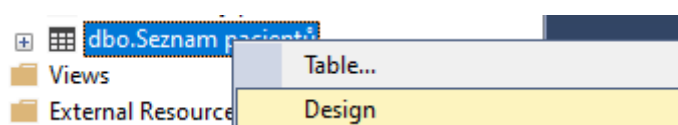


Obrázek 6 – Vybrání tabulek pro import

- g. V dalších krocích, pokud je již možné, se zvolí možnost Finish.
- h. Mělo by se zobrazit dialogové okno, že import proběhl v pořádku. Pokud je někde nějaký problém, vypíše se a muselo by se postupovat podle instrukcí.


Nyní bude ukázáno, jak se nastaví kardinalita mezi tabulkami.

- a. Nejdříve se musí nastavit primární klíče v tabulkách.
  - a. Klikneme pravým tlačítkem na tabulku a zvolíme design tabulky



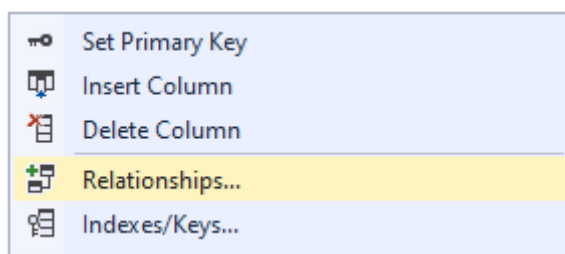
Obrázek 7 – Design tabulky

- b. V Designu tabulky klikneme pravým tlačítkem na příslušné pole a zvolíme Set Primary Key, změny uložíme, např. při zavření tabulky.

	Column Name	Data Type	Allow Nulls
	ID	float	<input type="checkbox"/>
	Jméno	nvarchar(30)	<input type="checkbox"/>

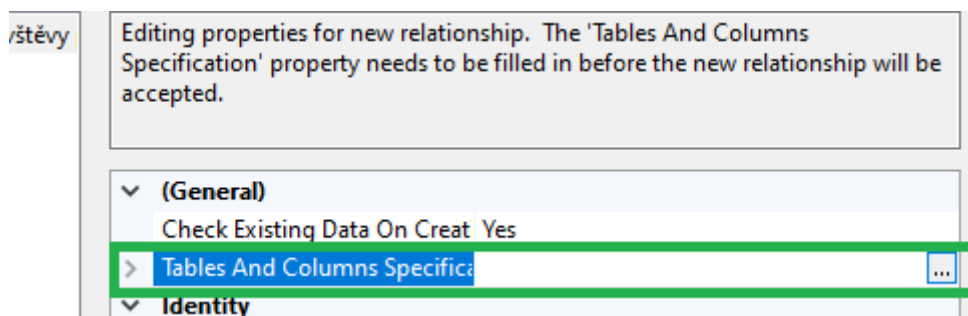
Obrázek 8 – Nastavení primárního klíče

- b. Nyní nastavíme kardinalitu mezi tabulkami. Je nutné kliknout na podřízenou tabulku, tj. tu, kde bude cizí klíč. Máme tedy tabulku otevřenou v designu jako v předcházejícím kroku. Když máme tabulku otevřenou, klikneme pravým tlačítkem myši kamkoliv do bílé části okna a vybereme možnost Relationships...



Obrázek 9 – Nastavení kardinality vztahu mezi tabulkami

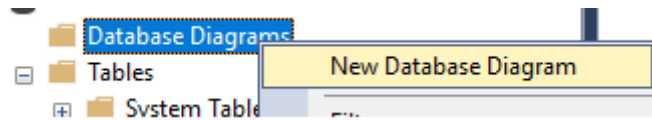
- c. V příslušném okně zvolíme možnost Add. Zvolíme druhou možnost a klikneme na 3 tečky, viz obrázek.



Obrázek 10 - Nastavení kardinality vztahu mezi tabulkami

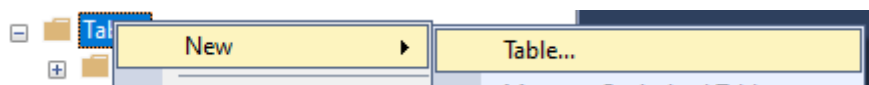
- d. V následujícím dialogovém okně můžeme vybrat pouze propojení primárního klíče (tabulku a následně pole), ovšem nemůžeme vybrat tabulku, která bude podřízená, bude v ní tzv. cizí klíč, proto jsme volili zobrazení podřízené tabulky s cizím klíčem. Nastavíme správně relaci mezi tabulkami, tj. vybereme pod tabulkami příslušné správné atributy. Potvrdíme Ok. Ještě můžeme relaci pojmenovat v Relationship name.
- e. Pokud bychom chtěli v dané tabulce vytvořit další kardinalitu s další tabulkou, museli bychom po vybrání možnosti Relationships... zvolit opět Add a postupovat stejně podle kroků uvedených výše.

- i. Pokud bychom si chtěli zobrazit relaci, kterou jsme vytvořili, klikneme na Database Diagrams a vybereme nový. Pokud by to nešlo a zobrazovala by se nám nějaká chyba, tak Management studio zavřeme, otevřeme, případně ještě zkusíme Refresh. Dále přidáme příslušné tabulky.



Obrázek 11 – Vytvoření relace mezi tabulkami

3. Pomocí Management Studia mohou žáci vytvořit tabulku i v návrhovém zobrazení, stejně jako v programu MS ACCESS. Možnost je uvedena níže na obrázku.



Obrázek 12 – Vytvoření tabulky v návrhovém zobrazení

Výhoda tohoto postupu je taková, že žáci si mohou zobrazit, jaké datové typy mohou používat, viz následující obrázek. Tj. třeba si mohou zkontrolovat, jestli mají navržené datové typy správně, případně jaké datové typy mají na výběr.

Column Name	Data Type	Allow Nulls
	<input type="text"/> bigint <b>binary(50)</b> bit char(10) date	<input type="checkbox"/>

Obrázek 13 - Vytvoření tabulky v návrhovém zobrazení

V této části bude primárně ukázáno vytvoření filmové databáze pomocí příkazů DDL jazyku SQL. Pro inspiraci žáci mohou využít např. následující webové stránky, které obsahují spoustu informací a příkladů, <https://www.w3schools.com/>.

Pro vytvoření nového dotazu se v SQL Serveru Management Studiu klikne na New Query a napíše se příslušný kód. Důležité při vytváření kardinality vztahu mezi tabulkami je přidat do příslušné tabulky atribut cizího klíče, pomocí **CONSTRAINT** pojmenovat kardinalitu vztahu. Pozor, i v různých tabulkách musí být tento název odlišný. Nemohu ve více tabulkách pojmenovat kardinalitu vztahu stejným názvem. Pomocí **FOREIGN KEY** určit, jaký atribut je cizím klíčem. A dále pomocí **REFERENCES** se odkázat na tabulku, ve které je cizí klíč primárním klíčem. Níže jsou uvedeny obrázky možného návržení daných tabulek. Nejdříve je nadefinovaná tabulka Film a dále jsou nadefinované další tabulky, které s ní souvisejí. Z popisu obrázků by snad mělo být vše jasné. Ve vazebních tabulkách kardinality vztahu M:N, oproti návrhu v předcházejícím příspěvku, jsou uvedeny primární klíče.

```

)CREATE TABLE Film
(
    [Id filmu] int NOT NULL PRIMARY KEY,
    [Název filmu] nvarchar(50),
    [Premiéra] date,
    [Rozpočet] money,
    [Tržby] money
);

```

Obrázek 14 – Vytvoření tabulky Film pomocí jazyku SQL

```

CREATE TABLE Produkce
(
    [Id produkce] int NOT NULL PRIMARY KEY,
    [Název produkce] nvarchar(50)
);

CREATE TABLE Film_Produkce
(
    [Id film-produkce] int NOT NULL PRIMARY KEY,
    [Id filmu] int,
    [Id produkce] int,
    CONSTRAINT FK_Film_Film_Produkce FOREIGN KEY ([Id filmu])
    REFERENCES [dbo].[Film]([Id filmu]),
    CONSTRAINT FK_Produkce_Film_Produkce FOREIGN KEY ([Id produkce])
    REFERENCES [dbo].[Produkce]([Id produkce])
);

```

Obrázek 15 – Vytvoření kardinality M:N mezi tabulkou Film a tabulkou Produkce.

```

CREATE TABLE Žánry
(
    [Id žánru] int NOT NULL PRIMARY KEY,
    [Název žánru] nvarchar(50)
);

CREATE TABLE Žánry_ve_filmu
(
    [Id film-žánry] int NOT NULL PRIMARY KEY,
    [Id filmu] int,
    [Id žánru] int,
    CONSTRAINT FK_Film_Žánry_ve_filmu FOREIGN KEY ([Id filmu])
    REFERENCES [dbo].[Film]([Id filmu]),
    CONSTRAINT FK_Žánry_Žánry_ve_filmu FOREIGN KEY ([Id žánru])
    REFERENCES [dbo].[Žánry]([Id žánru])
);

```

Obrázek 16 - Vytvoření kardinality M:N mezi tabulkou Film a tabulkou Žánry

```

)CREATE TABLE Národnost
(
    [Id národnosti] int NOT NULL PRIMARY KEY,
    [Národnost] nvarchar(50)
);

```

Obrázek 17 - Vytvoření tabulky Národnost pomocí jazyku SQL

```
CREATE TABLE Herec
(
    [Id herce] int NOT NULL PRIMARY KEY,
    [Jméno] nvarchar(50),
    [Příjmení] nvarchar(50),
    [Datum narození] date,
    [Poznámka] nvarchar(100),
    [Id národnosti] int,
    CONSTRAINT FK_Herec_Národnost FOREIGN KEY ([Id národnosti])
    REFERENCES [dbo].[Národnost]([Id národnosti])
);

CREATE TABLE Herec_hraje_ve_filmu
(
    [Id herec-film] int NOT NULL PRIMARY KEY,
    [Id filmu] int,
    [Id herce] int,
    [Role] nvarchar(50),
    CONSTRAINT FK_Film_Herec_hraje_ve_filmu FOREIGN KEY ([Id filmu])
    REFERENCES [dbo].[Film]([Id filmu]),
    CONSTRAINT FK_Herec_Herec_hraje_ve_filmu FOREIGN KEY ([Id herce])
    REFERENCES [dbo].[Herec]([Id herce])
);
```

Obrázek 18 - Vytvoření kardinality M:N mezi tabulkou Film a tabulkou Herec

```
CREATE TABLE Režisér
(
    [Id režiséra] int NOT NULL PRIMARY KEY,
    [Jméno] nvarchar(50),
    [Příjmení] nvarchar(50),
    [Id národnosti] int,
    CONSTRAINT FK_Režisér_Národnost FOREIGN KEY ([Id národnosti])
    REFERENCES [dbo].[Národnost]([Id národnosti])
);

CREATE TABLE Režisér_režiruje_film
(
    [Id režisér-film] int NOT NULL PRIMARY KEY,
    [Id filmu] int,
    [Id režiséra] int,
    CONSTRAINT FK_Film_Režisér_režiruje_film FOREIGN KEY ([Id filmu])
    REFERENCES [dbo].[Film]([Id filmu]),
    CONSTRAINT FK_Režisér_Režisér_režiruje_film FOREIGN KEY ([Id režiséra])
    REFERENCES [dbo].[Režisér]([Id režiséra])
);
```

Obrázek 19 - Vytvoření kardinality M:N mezi tabulkou Film a tabulkou Režisér



```

CREATE TABLE Recenzent
(
    [Id recenzenta] int NOT NULL PRIMARY KEY,
    [Jméno] nvarchar(50),
    [Příjmení] nvarchar(50),
    [Přezdívk] nvarchar(50),
    [Id národnosti] int,
    CONSTRAINT FK_Recenzent_Národnost FOREIGN KEY ([Id národnosti])
    REFERENCES [dbo].[Národnost]([Id národnosti])
);

CREATE TABLE Recenze_filmů
(
    [Id recenzent-film] int NOT NULL PRIMARY KEY,
    [Id filmu] int,
    [Id recenzenta] int,
    CONSTRAINT FK_Film_Recenze_filmů FOREIGN KEY ([Id filmu])
    REFERENCES [dbo].[Film]([Id filmu]),
    CONSTRAINT FK_Recenzent_Recenze_filmů FOREIGN KEY ([Id recenzenta])
    REFERENCES [dbo].[Recenzent]([Id recenzenta])
);

```

Obrázek 20 - Vytvoření kardinality M:N mezi tabulkou Film a tabulkou Recenzent

### Organizace práce

1. Žáci instalují SQL Server a SQL Server Management Studio.
2. Žáci importují filmovou databázi z prostředí MS ACCESS na SQL Server pomocí SQL Server Management Studia. Vytváří kardinality mezi tabulkami.
3. Žáci vytváří filmovou databázi pomocí jazyku SQL.

### Na co si dát pozor

- Občas pomůže zvolení možnosti Refresh. Případně, kdyby nešla nějaká nastavení, z důvodu rozsahu zde nebudou uvedena, stačí se obvykle z SQL Server Management Studio odhlásit a přihlásit znovu.

### Alternativní řešení

- Dalším prostředím, kde je možné danou filmovou databázi vytvořit, je alternativa k SQL Serveru a Management Studiu: databáze MySQL k SQL Serveru a software phpMyAdmin k Management Studiu.

### Zdroje:

- Snímky obrazovek: autor Ondřej Kořínek; zdroj vlastní, prostředí Microsoft Access 2019 a SQL Server Management Studio