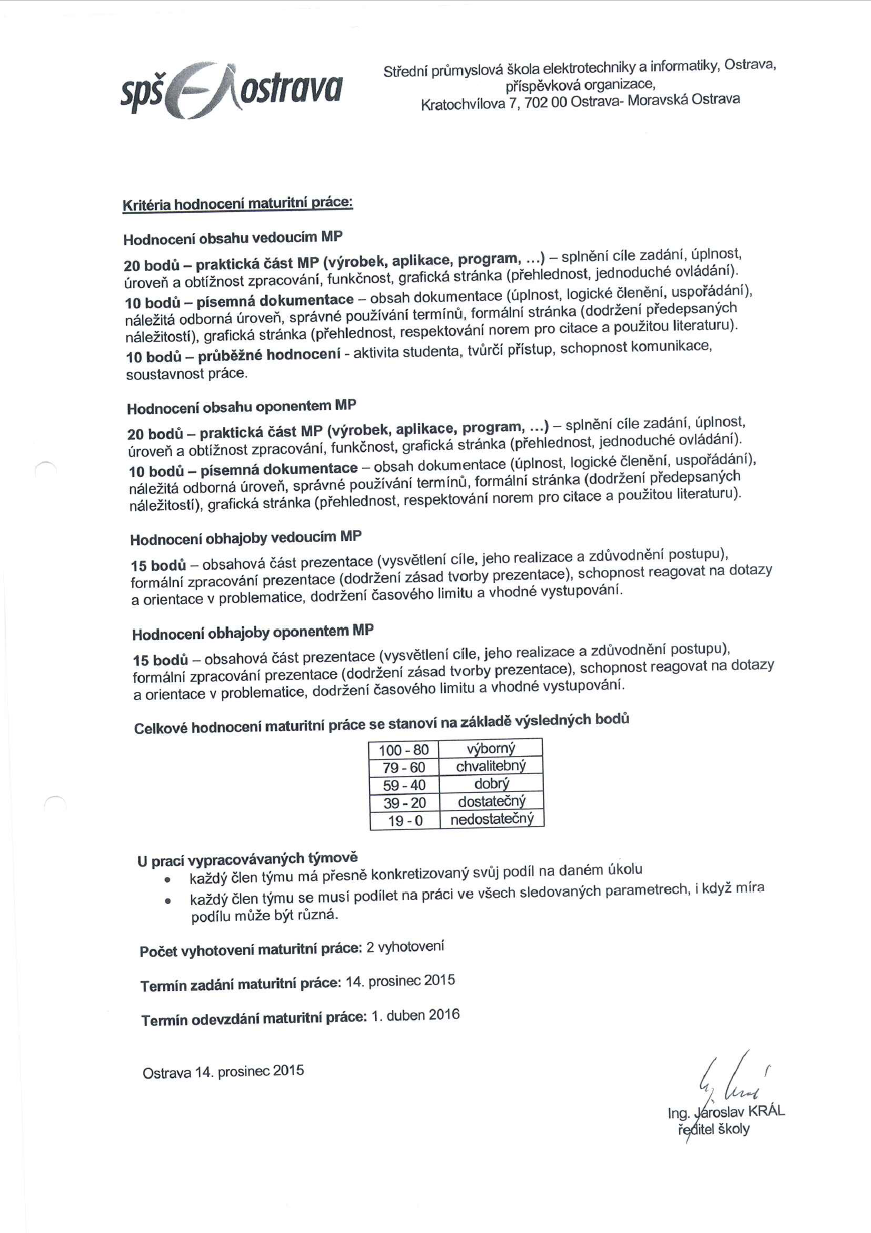
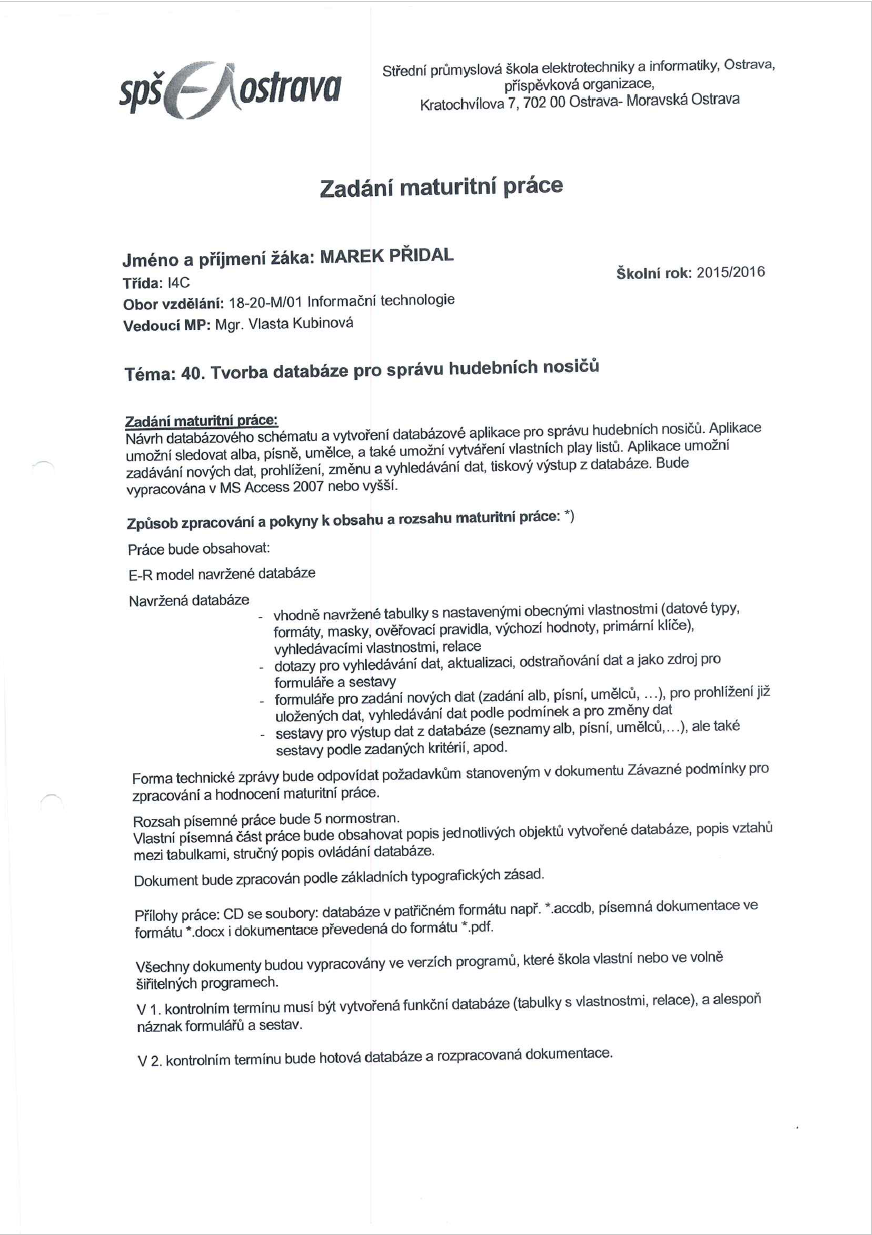
STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNIKY A INFORMATIKY OSTRAVA, PŘÍSPĚVKOVÁ ORGANIZACE

###### Databáze pro správu hudebních nosičů

Zpracoval: Marek Přidal Třída: I4C

Vedoucí práce: Mgr. Vlasta Kubinová Školní rok: 2015/2016



# Abstrakt

Databáze umožňuje jednoduchou správu vlastní nebo komerční hudební knihovny. Aplikace celkem pracuje s osmi tabulkami včetně tří propojovacích. Práce s databází je zajištěna pomocí formulářů vytvořených v Microsoft Access 2016. Kompatibilita je však zajištěna až do verze 2007. V rámci aplikace jsou i SQL soubory pro vytvoření tabulek a vztahů v jakémkoliv jiném programu pracující s SQL nebo MySQL.

Prohlašuji, že předložená práce je mým původním dílem, které jsem vypracoval samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.

Ostrava

podpis

Beru na vědomí, že tato maturitní práce je majetkem Střední průmyslové školy elektrotechniky a informatiky, Ostrava, příspěvková organizace (autorský zákon č. 121/2000 Sb., §60 odst. 1), bez jejího souhlasu nesmí být nic z obsahu publikováno.

Souhlasím / Nesouhlasím s prezenčním zpřístupněním své práce ve Školním informačním centru Střední průmyslové školy, Ostrava, příspěvková organizace.

Ostrava

podpis

Obsah

[1. Abstrakt 4](#_Toc445923427)

[2. Úvod 7](#_Toc445923428)

[3. E-R model 8](#_Toc445923429)

[4. Tabulky 10](#_Toc445923430)

[5. Dotazy 11](#_Toc445923431)

[6. Formuláře 13](#_Toc445923432)

[7. Sestavy 17](#_Toc445923433)

[8. Závěr 18](#_Toc445923434)

# Úvod

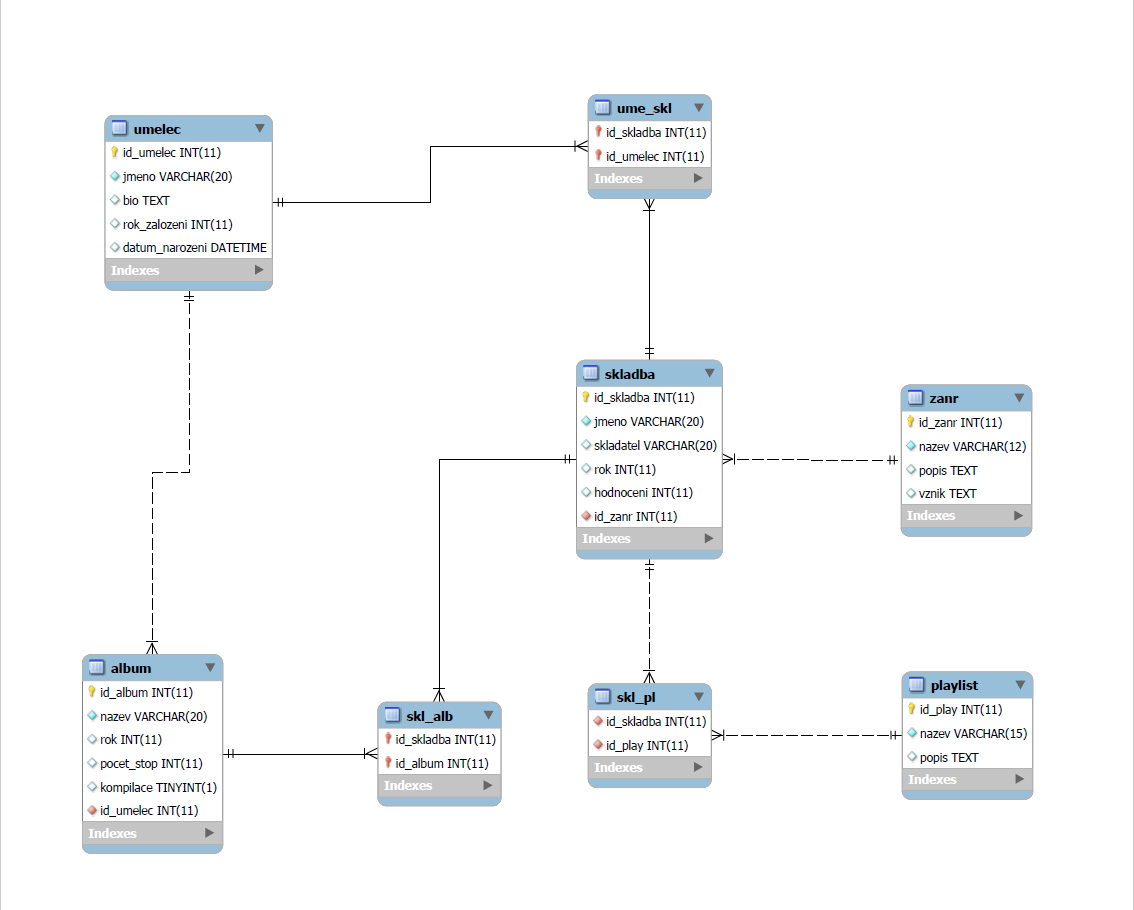
Jako téma dlouhodobé maturitní práce jsem si vybral databázi hudebních nosičů a skladeb. Podobný projekt jsem tvořil už ve třetím ročníku do předmětu Webové aplikace, kde však byla databáze mnohem jednodušší a hlavní důraz se kladl na práci s PHP a interakci s MySQL serverem.

Pro dlouhodobou maturitní práci jsem zvolil prostředí Microsoft Access. Hlavním důvodem je relativně jednoduché tvoření formulářů a hlavní důraz je tak kladen na správný návrh a propojení příslušných tabulek.

Základní návrh databáze je však k dispozici také ve zdrojových SQL souborech a díky tomu je usnadněno její použití na webovém serveru a následná interakce pomocí PHP. Navíc jsem také mohl využít pokročilejší E-R model z IDE Oracle MySQL Workbench, který na rozdíl od Microsoft Access umí přímo v modelu zobrazit rovněž datové typy.

# E-R model

E-R model se využívá pro grafické znázornění vztahů a entit databáze.



Obrázek : ER model vytvořený pomocí MySQL Workbench

Jako hlavní a stěžejní entita je brána tabulka Skladba. Od té se následně odvíjí všechny vztahy a další entity databáze. Zvolil jsem tak především proto, že hudební databáze podle mě stojí především na skladbách, bez kterých by databáze neměla smysl.

Ke skladbě jsou pak napojeny další tabulky, které samotný záznam skladby rozvíjí. Jako povinné jsem zvolil album a umělce. Bez těchto povinných záznamů by totiž mohla databáze jednoduše ztratit na přehlednosti. Pro změnu jako nepovinný vztah vůči skladbě ještě existují entity Žánr a Playlist. U Playlistu je souvislost asi jasná, napříč tomu u žánru jsem takový vztah zvolil především proto, že ne vždy je u skladby žánr úplně jasný a třeba u historicky starších písniček by mohlo být určení až příliš obtížné.

Až na vztah s žánrem jsem u všech ostatních volil vztah M:N. Následné propojení je pak řešeno přímo ve formuláři. V E-R model se pro tento vztah používá propojovací tabulka. Ta je tvořena ze dvou cizích klíčů, které dohromady tvoří složený primární klíč. Konkrétně to můžu uvést na vztahu mezi Skladbou a Umělcem. Zde je použit vztah M:N, tedy pro propojení se využívá propojovací tabulka ume\_skl, jež je tvořena dvěma atributy – id\_skladba a id\_umelec. Ty jsou zároveň cizími klíči a tvoří tak složený primární klíč entity.

U žánru je vztah 1:N především proto, že písnička se většinou zařazuje do jednoho žánru, a kdyby náhodou korelovala mezi vícero, dá se proto vytvořit nový záznam v tabulce Žánr. Není to však tak časté a proto jsem raději zvolil vztah 1:N.

Mimo skladbu ještě existuje nepovinný vztah mezi umělcem a albem. Ten jsem přidal až o něco později, protože jsem poznal, že získávání umělce přes skladbu je zbytečně složité a zdlouhavé. Navíc jde o vztah 1:N, takže nebylo zapotřebí použít speciální formulář pro vzájemné propojení.

# Tabulky

# Dotazy

Microsoft Access přímo podporuje práci s SQL dotazy. Ty jsem používal především pro vypisování z tabulek pomocí klauzule SELECT ale také třeba pro vyhledávání či mazání.

Využít se také dá návrhové zobrazení pro tvorbu dotazu, které je součástí Microsoft Access. Díky tomu je možné vytvořit požadovaný dotaz i s částečnou znalostí SQL a IDE se postará o zbytek. Díky tomu je navíc možné vytvořit také příslušnou SQL syntaxi pro další použití mimo Microsoft Access.

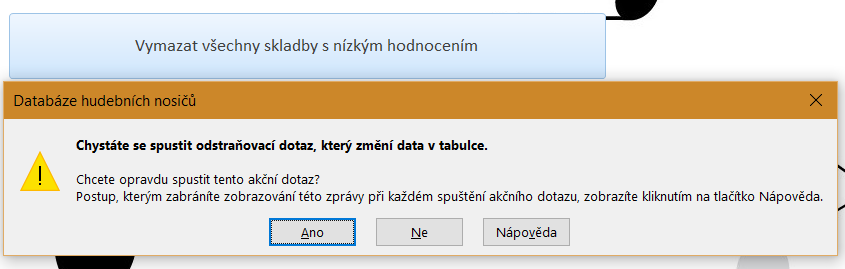
Microsoft Access také umí dotazy sám generovat na základě propojených tabulek, to se dá nejlépe využít primárně při tvorbě nekonečných formulářů, já jsem ale často volil cestu přes manuální SQL dotazy.

Součástí databáze je také jednoduché vyhledávání, které se nachází pod tlačítkem Vyhledávání v patičce každého nekonečného formuláře. Microsoft Access sice nepodporuje nativní práci s proměnnými, nabízí však tzv. parametry. Právě ten jsem využil při vyhledávání, kdy se do něj uloží zadaný název a následně se na novém formuláři zobrazí všechny záznamy, které v názvu nebo jménu splňují zadaný řetězec. V první verzi však funguje vyhledávání jen v případě, že uživatel zadá přesný název a zatím nepracuje tak chytře, aby dokázal skladby zobrazit i na základě prvního písmena. To bych však zařadil až do dalších verzí vzhledem k omezenému času, který jsem pro práci na databázi měl.



Obrázek : Vyhledávání

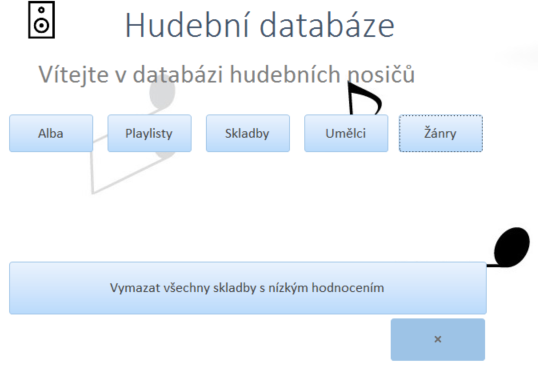
Druhým užitečným dotazem je funkce pro odstranění všech skladeb, které splňují kritérium a disponují hodnocením menším než 2. To se může hodit především ve chvíli, kdy už v databázi je až příliš moc záznamů a uživatel by rád vymazal ty nejméně oblíbené. Od toho se odvíjí podmínka, kdy se předpokládá, že nejméně oblíbené skladby budou mít také nejnižší hodnocení. Ještě před vymazáním se databáze uživatele zeptá, jestli si je skutečně operací jist a taky se mu zobrazí, o kolik záznamů tímto krokem přijde.



Obrázek : Dotaz pro odstranění neoblíbených skladeb

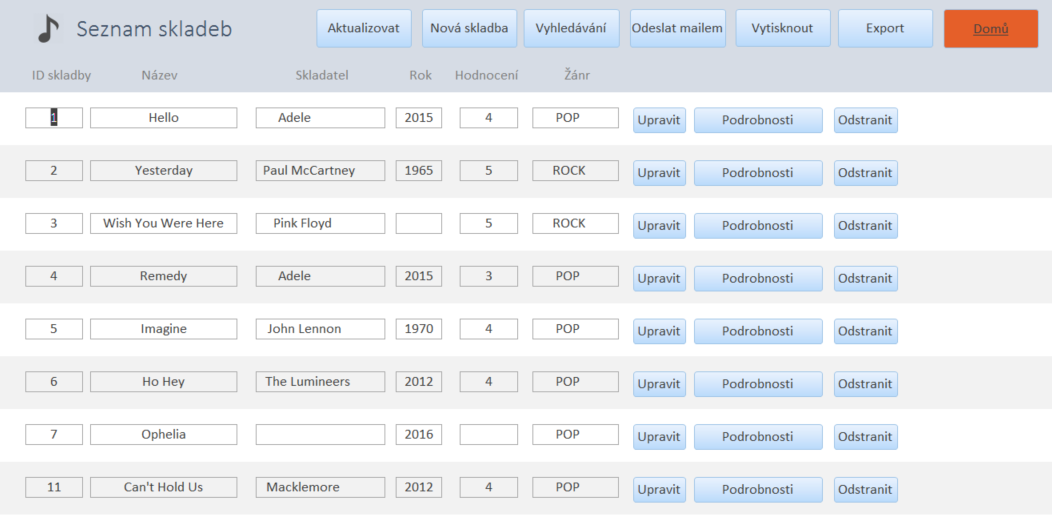
# Formuláře

Formuláře slouží k jednoduché komunikaci s databází. Jsou určeny především pro běžné uživatele, kteří nemají žádnou pokročilou znalost v oblasti databází a pomocí formulářů se jim zamezí, aby databázi zásadním způsobem narušili.



Obrázek : Úvodní formulář

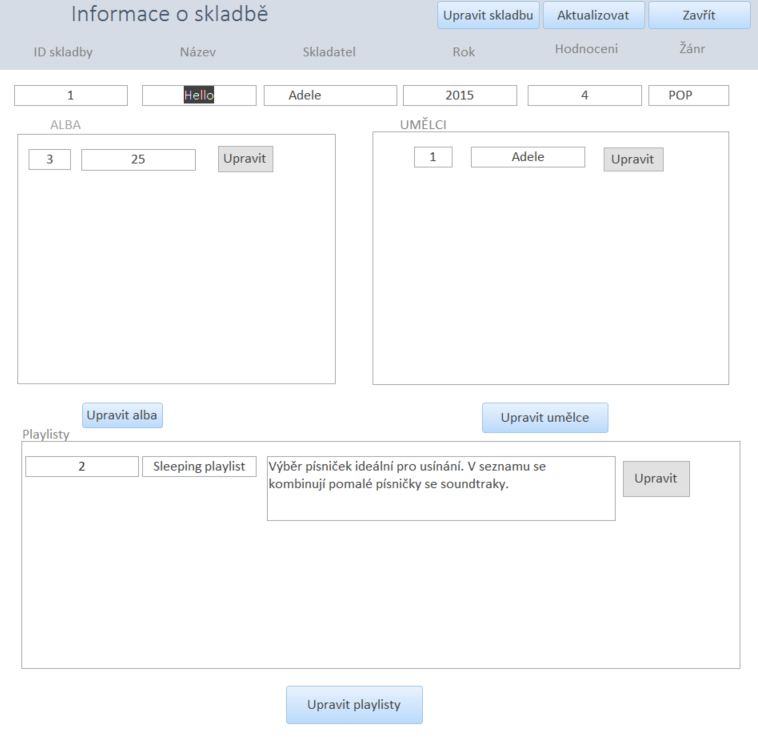
Můj úvodní formulář umožňuje vstup do pěti nekonečných formulářů vypisujících data z databáze. Krom toho ještě v dolní části nechybí možnost pro spuštění odstraňovacího dotazu pro smazání všech skladeb s nízkým hodnocením, tedy skladeb s hodnocením menším než 2. Černý křížek v pravém dolním rohu slouží k zavření databáze.



Obrázek : Seznam skladeb

Pro demonstraci jsem vybral nekonečný formulář Seznam skladeb, který je nejobsáhlejší. V horní části formuláře je pomocí tlačítek umožněno celý formulář aktualizovat, přidat nový záznam, vyhledávat mezi skladbami, odeslat celý seznam písniček mailem, vytisknout jej a exportovat. Tlačítko domů vrátí uživateli zpět na úvodní formulář a aktuální zavře.

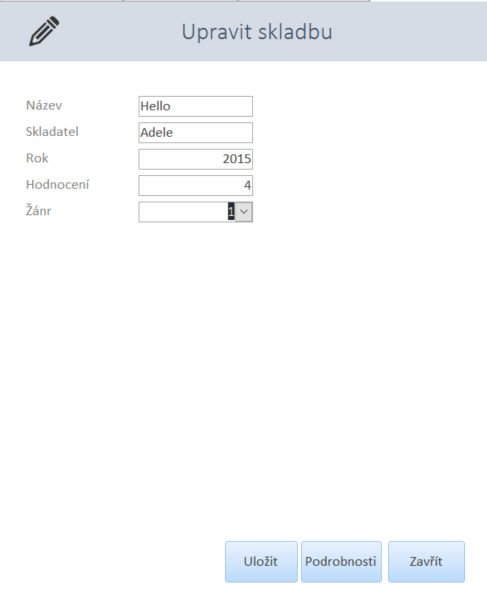
Přímo z nekonečného formuláře je možné záznamy upravit, odstranit nebo zobrazit podrobnosti.



Obrázek : Informace o skladbě

V informacích o skladbě využívám formulář, který je ještě složen ze tří dalších podformulářů. Ty slouží pro výpis záznamů, které jsou pro danou skladbu relevantní a nachází se v jiných tabulkách. Z tohoto formuláře je přímo možné upravit propojující vztah nebo propojený záznam. Ke všem těmto úpravám slouží příslušná tlačítka upravit u záznamů.

V horní části formuláře jsou tlačítka, která umožňují upravit konkrétní záznam dané skladby, případně celý formulář aktualizovat po provedené změně. Tlačítko zavřít formulář zavře a uživatele vrátí na předchozí otevřený formulář.



Obrázek : Upravit skladbu

Formulář pro úpravu skladby je vcelku jednoduchý a krom jednoduchého zadání příslušných atributů umožňuje také propojení s požadovaným žánrem. Díky tomu, že mezi entitou Skladba a Žánr je vztah 1:N, není zapotřebí propojovací tabulky a díky správným vyhledávacím vlastnostem stačí příslušný žánr vybrat z rozbalovací nabídky.

# Sestavy

# Závěr