



FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Advokátní kancelář

KIV/DB1 – Semestrální práce

student:	<i>Tomáš Vitek</i>
osobní číslo:	<i>A21B031P</i>
email:	<i>twitty@students.zcu.cz</i>
datum:	<i>27.12.2022</i>

CHARAKTERISTIKA SEMESTRÁLNÍ PRÁCE

Cílem mojí semestrální práce bylo vytvořit funkční databázový model pro provoz advokátní kanceláře. Advokátní kancelář nabízí své služby klientům, s klienty o této činnosti sepisuje smlouvy a jejich průběh zaznamenává advokátních spisech. K tomu bylo použito entity: *smlouva*, *advokat*, *klient*, *poskytuje*, *sluzba* a *avokatni spis*.

KLIENT

Tabulka klient obsahuje údaje o konkrétních klientech advokátní kanceláře. Primární klíč klienta je uměle vytvořené *id_klienta*, které nemá žádný další význam. Kromě toho má každý klient atributy: *jmeno*, *prijmeni*, *datum_narozeni*, *telefon* a nepovinný atribut *email*. Datum narození klienta nesmí být v budoucnosti, v databázovém modelu je budoucnost pro zjednodušení chápána od 1.1.2023 výše.

Klient uzavírá smlouvu s advokátem na nějakou službu. Klient může uzavřít i více smluv. Smlouva je však uzavírána pouze s jedním klientem.

SMOUVA

Tato tabulka obsahuje smlouvy, které sjednává advokát s klientem. Předmětem smlouvy je poskytnutí služby klientovi. Průběh poskytování služby je pak popisován v advokátním spisu. Primárním klíčem je *cislo_smlouvy* fungující i jako její identifikátor. Atributem smlouvy jsou *datum_uzavreni* a *typ* představující název smlouvy. Cizími klíči jsou tedy: *id_klienta*, *id_advokata*, *id_sluzby*.

ADVOKÁT

Advokát má množinu služeb, které je schopen klientovi poskytnout. Na tyto služby může s klientem sjednat smlouvu. Mimo to je z administrativních důvodů schopen vést jakýkoliv, i nevlastní, advokátní spis.

Tabulka obsahuje primární klíč *id_advokata* a atributy *jmeno*, *prijmeni*, *titul* a *pocatek_praxe*, jehož hodnota je databází kontrolována. Mimo to je možné vložit i nepovinný atribut *specialize* představující právní odvětví.

SLUŽBA

Tabulka služeb představuje výčet produktů nabízených advokátní kanceláří. Tyto produkty pak poskytují jednotliví advokáti. Tabulka je tvořena primárním klíčem *id_sluzby*, který nemá žádný další využití. Dalšími povinnými atributy jsou *cena* a *typ*, představující název služby.

POSKYTUJE

Jedná se o rozkladovou tabulku vazby M:N, která by jinak vznikla mezi tabulkami služeb a advokátů. Advokát totiž může poskytovat více služeb a služba může být poskytována více advokáty.

Tato tabulka obsahuje pouze cizí klíče *id_sluzby* a *id_advokata*.

ADVOKÁTNÍ SPIS

Advokátní spis popisuje vykonávanou službu kardinalitou 1:1. Primárním klíčem je *cislo_spisu* fungující jako identifikátor spisu. Tabulka má jediný nenulový atribut *pocatek_vedeni_spisu* a celou řadu volitelných atributů *konec_vedeni_spisu*, *lhuty*, *dotcene_organy*, *ukony* a *poznámky*. Tabulka obsahuje cizí klíče *id_advokata* a *cislo_smlouvy*.

DATABÁZOVÉ POHLEDY

Pro zjednodušení práce s databází, jsem vytvořil dva základní pohledy.

První pohled *neuzavřené spisy* vypíše advokátní spisy, u kterých ještě nebyl vyplněn atribut *konec_vedeni_spisu*. Následně vypsány hodnoty atributů *cislo_spisu*, *sluzba* a *prijmeni* u klienta.

```
CREATE VIEW neuzavrene_spisy AS
SELECT `advokatni spis`.cislo_spisu AS "Číslo spisu", sluzba.typ AS "Služba", klient.prijmeni AS "Klient"
FROM sluzba, `advokatni spis`, smlouva, klient
WHERE `advokatni spis`.konec_vedeni_spisu IS NULL AND
`advokatni spis`.cislo_smlouvy = smlouva.cislo_smlouvy AND
smlouva.id_sluzby = sluzba.id_sluzby AND
smlouva.id_klienta = klient.id_klienta
ORDER BY `advokatni spis`.pocatek_vedeni_spisu;
```

Tento pohled při nezměněné databázi vrací výstup:

Číslo spisu	Služba	Klient
2	Zastoupení	Bezradný
1	Právní analýza	Da Vinci
5	Právní analýza	Bezradný

Druhý pohled s názvem *klienti advokátů* slouží k vypsání seznamu advokátů a jejich klientů, kterým je nebo byla poskytována nějaká služba.

```
CREATE VIEW klienti_advokatu AS
SELECT distinct concat (advokat.jmeno, ' ', advokat.prijmeni) AS "Advokát", concat(klient.jmeno, ' ', klient.prijmeni) AS "Klient"
FROM klient, advokat, smlouva
WHERE klient.id_klienta = smlouva.id_klienta AND advokat.id_advokata = smlouva.id_advokata
ORDER BY advokat.id_advokata;
```

Tento pohled při nezměněné databázi vrací výstup:

Advokát	Klient
Karel Novák	Lukáš Daněk
Karel Novák	Tomáš Bezradný
Bedřich Král	Leonardo Da Vinci
Bartoloměj Král	Lukáš Daněk

TESTOVACÍ DATA

Správnou funkci databáze ověřím pomocí dvou sérií dotazů a výpisů před a po úpravách.

V prvním testovacím scénáři prvně mimo jiné provedeme:

Přidáme nového advokáta:

```
INSERT INTO advokat (`id_advokata`, `jmeno`, `prijmeni`, `titul`, `pocatek_praxe`, `specializace`) VALUE (NULL, 'Čestmír', 'Kutnohorský', 'JUDr., Ph.D', '2008-04-15', NULL);
```

Přidáme novému advokátovi služby, které může poskytovat:

```
INSERT INTO poskytuje(`id_advokata`,`id_sluzby`) VALUES (5,1), (5,2), (5,3), (5,4);
```

Nyní zkusíme vytvořit smlouvu na službu, kterou advokát nedokáže poskytovat:

```
INSERT INTO smlouva (`cislo_smlouvy`, `typ`, `id_advokata`, `id_klienta`, `id_sluzby`, `datum_uzavreni`) VALUE (6, "Poskytnutí právních služeb", 5, 2, 5, "2021-10-12");
```

Toto nám databáze nedovolí, proto změníme služby, které může advokát poskytovat:

```
UPDATE `poskytuje` SET `id_advokata`=5,`id_sluzby`=5 WHERE `id_advokata`=5 AND `id_sluzby`=4;
```

A znovu zkusíme sjednat danou smlouvu:

```
INSERT INTO smlouva (`cislo_smlouvy`, `typ`, `id_advokata`, `id_klienta`, `id_sluzby`, `datum_uzavreni`) VALUE (6, "Poskytnutí právních služeb", 5, 2, 5, "2021-10-12");
```

Toto již je platný zápis.

Ve druhém testovacím scénáři mimo jiné vyzkoušíme integritní omezení databáze.

Zkusíme tak přidat advokátní ke smlouvě, která v databázi není.

```
INSERT INTO `advokatni spis` (`cislo_spisu`, `id_advokata`, `cislo_smlouvy`, `dotcene_organu`, `lhuty`,`ukony`, `poznámky`, `pocatek_vedeni_spisu`, `konec_vedeni_spisu`) VALUE (NULL, 3, 32, NULL, NULL, NULL, NULL, "2022-04-01", "2022-08-11");
```

Toto nám databáze neumožní vložit. Advokátní spis může být veden pouze pro existující smlouvy.

Nyní se pokusíme do databáze vložit smlouvu s neexistujícím klientem.

```
INSERT INTO smlouva (`cislo_smlouvy`, `typ`, `id_advokata`, `id_klienta`, `id_sluzby`, `datum_uzavreni`) VALUE (null, 'Poskytnutí právních služeb', 1, 24, 4,'2022-06-01');
```

Hodnota nebude do databáze zapsána,

Nyní zkusíme do hodnoty zapsat klienta s datem narození v budoucnosti.

```
INSERT INTO klient (`id_klienta`, `jmeno`, `prijmeni`,`datum_narozeni`, `telefon`,`e-mail`) VALUE (NULL, "Petr", "Brzorodý", '2052-04-11', 98756789, NULL);
```

Nelze.

Podobně nebude možné vložit do databáze nesmyslné hodnoty pro počátek praxe advokáta.

Počátek praxe nesmí být příliš starý ani se nesmí odehrávat v budoucnosti.

```
INSERT INTO advokat (`id_advokata`, `jmeno`, `prijmeni`, `titul`, `pocatek_praxe`, `specializace`) VALUE (null, "Vítězslav", "Zkušný", "JUDr", '1910-01-12', "Norimberské zákony");
```

```
INSERT INTO advokat (`id_advokata`, `jmeno`, `prijmeni`, `titul`, `pocatek_praxe`, `specializace`) VALUE (null, "Smil", "Podivný", "JUDr",'2142-01-12', "Newtonův druhý zákon");
```

ZÁVĚR

Ačkoliv je databáze vůči reálné situaci výrazně zjednodušená, podařilo se mi ji napsat plně funkční a schopnou splnit požadované vlastnosti. Pro lepší přehlednost a užívání by však bylo vhodné napsat aplikaci ovládající databázi.