

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

MODELOVANIE RELAČNÝCH DATABÁZ V PRÍKLADOCH
BAKALÁRSKA PRÁCA

2015

MICHAL NEMEŠ

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

MODELOVANIE RELAČNÝCH DATABÁZ V PRÍKLADOCH
BAKALÁRSKA PRÁCA

Študijný odbor: 9.2.9 Aplikovaná informatika
Študijný program: Aplikovaná informatika
Školiace pracovisko: Katedra informatiky
Školiteľ: Mgr. Martin Drlík, PhD.

Nitra 2015

Michal Nemeš



Univerzita Konštantína Filozofa v Nitre
Fakulta prírodných vied

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Michal Nemeš
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.9 aplikovaná informatika
Typ záverečnej práce: Bakalárska práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Modelovanie relačných databáz v príkladoch

Anotácia: Moderné databázové systémy sú súčasťou širokého spektra súčasných aplikácií a systémov. Rozumieť princípom návrhu relačných databáz predstavuje jednu z hlavných požiadaviek na vedomosti absolventov aplikovanej informatiky.
Ciele bakalárskej práce:
Po oboznámení sa s odporúčanými postupmi návrhu relačných databáz študent vytvorí podporný študijný materiál, ktorý bude obsahovať sadu riešených príkladov z oblasti návrhu relačných databáz. Študent vo vybranom nástroji navrhne väčší počet databázových návrhov, na ktorých odprezentuje overené postupy, ako aj typické problémy, s ktorými sa počas jednotlivých etáp návrhu možno stretnúť.
Požiadavky na vedomosti a zručnosti:
angličtina, skúsenosti s návrhom databáz

Školiteľ: Mgr. Martin Drlík, PhD.
Oponent: RNDr. Ján Skalka, PhD.
Katedra: KI - Katedra informatiky
Dátum zadania: 02.10.2013

Dátum schválenia: 17.10.2013

prof. Ing. Milan Turčáni, CSc.
schválil/a

POĎAKOVANIE

Chcel by som sa touto cestou poďakovať môjmu školiteľovi Mgr. Martinovi Drlíkovi, PhD., za cenné pripomienky a odborné vedenie pri tvorbe bakalárskej práce.

ABSTRAKT

NEMEŠ, Michal: Modelovanie relačných databáz v príkladoch. [Bakalárska práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied. Školiteľ: Mgr. Martin Drlík, PhD. Stupeň odbornej kvalifikácie: Bakalár odboru Aplikovaná informatika. Nitra: FPV, 2015. 49 s.

Problematikou modelovania relačných databáz sa v súčasnosti zaoberá stále viac ľudí pracujúcich v oblasti informačných technológií. V bakalárskej práci sú obsiahnuté informácie o databázach a metodike návrhu relačných databáz. Práca sa delí na teoretickú a praktickú časť. Teoretická časť zahŕňa analýzu súčasného stavu vývoja databázových systémov a ich využitie v praxi. Súčasťou je tiež popis dátových modelov, porovnanie relačných a nerelačných databáz a popis systémov riadenia bázy dát. Teoretickú časť uzatvára metodika návrhu relačných databáz s informáciami o jednotlivých súčiastiach relačného modelu. V tejto časti bakalárskej práce sa nachádza zhrnutie, kde sú obsiahnuté metódy predchádzania vzniku možných problémov pri spracúvaní databáz. Praktická časť sa zameriava na tvorbu konkrétnych databázových modelov, s využitím odporúčaných postupov pre návrh modelu databázy. Bakalárska práca by mohla slúžiť ako podporný študijný materiál pre čitateľov, ktorí začínajú s návrhom relačných databáz.

Kľúčové slová: Databáza. Metodika. Relačný model. Návrh modelov.

ABSTRACT

NEMEŠ, Michal: Relation Data Modelling By Examples. [Bachelor Thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences. Supervisor: Mgr. Martin Drlík, PhD. Degree of Qualification: Bachelor of Applied Informatics. Nitra: FNS, 2015. 49 p.

Issue of relational database modeling is nowadays increasingly popular among people working in information technology. Bachelor thesis contains information about databases and methodology of relational databases design. Thesis is divided into theoretical and practical part. Theoretical part includes fundamental terms and definitions. It incorporates explanation of the term database and states examples of its uses in practice. Description of data models, comparison of relational and non-relational databases and description of various database management systems are also included in this part. Theoretical part is concluded with a methodology of relational database design and information about individual components of a relational model. This part of bachelor thesis contains summary of methods used to prevent possible problems with handling the database. Practical part is focused on creating particular database models with use of recommended methods of relational database modeling. Bachelor thesis could serve as learning support material for readers who are starting to learn relational data modeling.

Keywords: Database. Methodology. Relational model. Database design.

OBSAH

Úvod	9
1 Analýza súčasného stavu.....	10
1.1 Využitie databáz	10
1.2 Systémy riadenia bázy dát	10
1.3 Relačné a nerelačné databázy	12
1.4 Dátové modely	13
1.5 Entitno-relačný model	14
1.6 CASE nástroje	16
1.7 Dátová integrita	16
1.8 Metodika návrhu relačných databáz	17
1.8.1 Vytvorenie tabuliek	17
1.8.2 Definovanie atribútov	17
1.8.3 Určenie kľúčov	19
1.8.4 Definovanie obmedzení	20
1.8.5 Vytvorenie vzťahov medzi tabuľkami	21
1.8.6 Normalizácia návrhu	23
2 Ciele bakalárskej práce.....	25
3 Zbierka príkladov.....	26
3.1 Filmová databáza	27
3.2 Rezervácie športovísk	30
3.3 Evidencia knižnice	31
3.4 Databáza mobilných telefónov	32
3.5 Databáza pracovných ponúk	33
3.6 Rezervácie kúpeľov, ich služieb a hotelov	34
3.7 Reštaurácia	35
3.8 Hokejová liga	36
3.9 Práca na projektoch	37
3.10 Predaj vstupeniek do kina	38
3.11 Požičovňa automobilov	39
3.12 Autoškola	40
3.13 Nemocnica	41
3.14 Evidencia Olympijských hier	42
3.15 Ubytovanie v apartmánoch	43
3.16 Psí útulok	44

3.17	Eshop.....	45
Záver		47
Zoznam bibliografických odkazov		48
Zoznam príloh		50

ZOZNAM ILUSTRÁCIÍ A TABULIEK

Obr. 1 a) Nesprávne definovanie atribútov, b) Správne definovanie atribútov	19
Obr. 2 Znázornenie vzťahu 1:1	21
Obr. 3 Znázornenie vzťahu 1:n	21
Obr. 4 Znázornenie vzťahu m:n	22
Obr. 5 Znázornenie voliteľností účasti na vzťahu	23
Obr. 6 Druhá normálna forma je: a) nesplnená, b) splnená	24
Obr. 7 Tretia normálna forma je: a) nesplnená, b) splnená	24
Obr. 8 Diagram aktivít procesu návrhu databázy	27
Obr. 9 Dátový návrh filmovej databázy	29
Obr. 10 Dátový návrh rezervácií športovísk	30
Obr. 11 Dátový návrh knižnice	31
Obr. 12 Dátový návrh databázy mobilných telefónov	32
Obr. 13 Dátový návrh databázy pracovných ponúk	33
Obr. 14 Dátový návrh rezervácií služieb kúpeľov a hotelových izieb	34
Obr. 15 Dátový návrh reštaurácie	35
Obr. 16 Dátový návrh pre evidenciu hokejovej ligy	36
Obr. 17 Dátový návrh práce na projektoch	37
Obr. 18 Dátový návrh predaju vstupeniek do kina	38
Obr. 19 Dátový návrh požičovne automobilov	39
Obr. 20 Dátový návrh evidencie študentov autoškoly	40
Obr. 21 Dátový návrh nemocnice	41
Obr. 22 Dátový návrh evidencie olympijských hier	42
Obr. 23 Dátový návrh ubytovaní v apartmánach	43
Obr. 24 Dátový návrh psieho útulku	44
Obr. 25 Dátový návrh eshopu	46

ÚVOD

Ľudia si v súčasnosti postupne zvykajú na to, že im moderné technológie pomáhajú uľahčiť život. Neodmysliteľnou súčasťou sa stáva spracovávanie informácií. Informácie majú totiž pre ľudí veľkú hodnotu už od počiatku vekov. V dnešnej dobe je dôležitým aspektom práve práca s dátami a ich ukladanie. Existujú prostriedky, vďaka ktorým je možné zhromažďovať dáta týkajúce sa určitej problematiky. Takéto sady dát sa nazývajú databázy.

Databázy sa využívajú v každodennom živote pri evidovaní dát užitočných pre spoločnosti i jednotlivcov. Slúžia na systematické ukladanie dát, vďaka čomu je práca s týmito dátami prehľadnejšia a rýchlejšia. Navyše, pri správne navrhnutej databáze sa predchádza zbytočnému opakovaniu dát.

Cieľom bakalárskej práce je popísať metodiku návrhu relačných databáz a vytvoriť zbierku riešených príkladov. Prvá kapitola je zameraná na oboznámenie so základnými pojmami a zahŕňa tiež vhodný postup návrhu relačných databáz, ako aj chyby, ktoré sa počas tohto procesu môžu vyskytnúť. Súčasťou je i riešenie vzniknutých anomálií prostredníctvom postupu nazývaného normalizácia. V druhej kapitole sú zhrnuté ciele bakalárskej práce. Ďalšiu kapitolu tvorí zbierka riešených príkladov, ktoré majú čitateľovi pomôcť uplatniť teoretické vedomosti nadobudnuté v predchádzajúcich kapitolách pri navrhovaní vlastných databáz. Príklady sú zoradené podľa náročnosti od jednoduchších k zložitejším, pričom prvý príklad obsahuje podrobný postup návrhu filmovej databázy. Ak sa v príkladoch vyskytnú situácie, ktoré môžu byť čitateľovi nejasné, alebo si zaslúžia špeciálnu pozornosť, sú uvedené v pomôcke nad riešením daného príkladu. Na vizuálne zobrazenie riešenia príkladov bol použitý program MySQL Workbench, ktorý je prispôsobený na vytváranie návrhov databáz v podobe entitno-relačného diagramu.

Vytváranie návrhov databáz môže pomôcť jednotlivcom pri organizovaní vlastných dát, ale je tiež dobre uplatniteľné na trhu práce, kde je stále viac ponúk pre ľudí pracujúcich s databázami.

1 ANALÝZA SÚČASNÉHO STAVU

Databázy sú sady vzájomne súvisiacich dátových položiek, ktoré sú spravované ako jediná položka. Výhodou databáz je, že umožňujú pracovať s veľkými dátovými sadami, pričom zabezpečujú ľahký prístup k údajom, umožňujú jednoduchosť vkladania nových dát do databázy, upravovania a mazania už existujúcich údajov v databáze. K jednej databáze môže súčasne pristupovať viacero používateľov, pričom je možné zamedziť prístup k niektorým dátam, prostredníctvom čoho sa zvyšuje bezpečnosť databázy. Údaje z databáz tiež môžu byť využívané inými aplikáciami (Sumathi, Esakkirajan, 2007; Oppel, 2006).

1.1 VYUŽITIE DATABÁZ

Vo sfére podnikania sa databázy využívajú nepretržite na spracúvanie objednávok, účtovníctvo, zaznamenávanie informácií o preprave. Obchody tiež uchovávajú v databáze informácie o zákazníkoch, vďaka čomu potom môžu zákazníkom zasielať špeciálne ponuky alebo ponuky so zľavami. Databázy sú tiež nevyhnutnou súčasťou hotelových recepcií, moderných knižníc, letísk, či bánk. V školstve sa už dnes väčšinou využívajú databázy na ukladanie informácií o učiteľoch, študentoch a ich známkach, ich dochádzke a ostatných študijných výsledkoch. Neziskové organizácie využívajú databázy na uchovávanie informácií o finančných príspevkoch, dobrovoľníkoch, klientoch ktorým organizácia pomohla a na ďalšie činnosti s tým súvisiace. V domácnostiach je možné používať databázy na ukladanie zbierky filmov alebo hudby, telefónnych čísiel a adries svojich známych (Connolly, Begg, 2005).

1.2 SYSTÉMY RIADENIA BÁZY DÁT

Systém riadenia bázy dát je softvér zabezpečujúci služby pre organizáciu a udržiavanie databázy. Umožňuje presúvanie dát do a z fyzických dátových súborov, prístup k dátam pomocou dotazovacieho jazyka, správu transakcií a správu simultánných prístupov viacerých používateľov k dátam. Systém riadenia bázy dát obsahuje zálohovacie mechanizmy pre zotavenie databázy v prípade havárie a bezpečnostné mechanizmy zabráňujúce neoprávnenému prístupu k dátam a neoprávneným modifikáciám. V súčasnosti je na trhu veľký výber databázových

systémov. Patria medzi ne MySQL, MariaDB, PostgreSQL, MongoDB a CouchDB a mnohé ďalšie (Oppel, 2012).

MySQL, vyvíjaný spoločnosťou Oracle, je open source systém riadenia bázy dát patriaci medzi najznámejšie systémy riadenia relačných databáz. Je to multiplatformový systém podporovaný všetkými hlavnými platformami, ako sú Windows, Linux, UNIX a ďalšie, pričom výkon databázy nie je závislý od použitej platformy. Ďalšou výhodou je, že rozhrania pre programovanie aplikácií umožňujú ľahkú spoluprácu s programovacími jazykmi ako sú Java, C, C++, Python, Perl a ďalšie. Menšie spoločnosti a bežní používatelia, najmä študenti, obľubujú tento systém aj kvôli tomu, že prevádzkové náklady sú nízke a existuje veľké množstvo bezplatných návodov na uľahčenie a zdokonalenie manipulácie s MySQL databázami (Gilmore, 2010).

MariaDB vznikla ako vetva MySQL, ktorá sa neskôr osamostatnila. Aj napriek oddeleniu si zachovala väčšinu funkcií. Zastáva pozíciu alternatívy MySQL, pričom prechod na MariaDB je jednoduchý, pretože súbory v ktorých sú databázy uchovávané, sú kompatibilné s oboma systémami (Forta, 2011).

PostgreSQL je systém rozvíjaný komunitou prispievateľov z celého sveta. Použitie, kopírovanie a upravovanie systému organizáciami vyžaduje iba oznámenie o autorských právach. PostgreSQL kladie dôraz na spoľahlivosť, vďaka čomu bezpečne zvláda dôležité aplikácie s vysokým počtom transakcií a veľkým počtom údajov a umožňuje dobrú kontrolu prístupu do databázy (Riggs, Krosing, 2010).

MongoDB je systém riadenia nerelačných databáz navrhnutý pre ukladanie a spracovanie veľkých dát. Nakoľko sa jedná o nerelačnú databázu, MongoDB nemá schému. Vďaka tomu je možné jednoducho upraviť, zväčšiť alebo pridať nové polia do dátového modelu, pričom nevyužívané polia nezaberajú žiadne miesto. MongoDB je vhodné riešenie pre databázy s veľkým objemom dát, no menej vhodné pre databázy, v ktorých sa očakáva vysoký počet transakcií (Membrey, Plugge, Hawkins, 2010).

CouchDB je systém riadenia nerelačných databáz vyvinutý s dôrazom na jednoduchosť používania. Zatiaľ čo väčšina databáz používa uzamykanie, aby nedošlo k súčasnej úprave dát tabuľky, CouchDB umožňuje čítanie aj počas editovania dát tabuľky. To vedie k efektívnejšiemu využívaniu zdrojov pri veľkých záťažach (Spoiala, 2013).

1.3 RELAČNÉ A NERELAČNÉ DATABÁZY

Známe sú relačné databázy a nerelačné databázy. Relačná databáza vychádza z relačného modelu, ktorý ukladá dáta do dvojrozsmernej tabuľky, kde stĺpce predstavujú atribúty tabuľky a riadky predstavujú samotné záznamy. Relačné databázy umožňujú vytvárať vzťahy medzi jednotlivými tabuľkami databázy, vďaka čomu je možné jednoducho pristupovať k dátam z viacerých tabuliek bez toho, aby najprv musela byť definovaná nejaká hierarchia. Vďaka tomu sú relačné databázy veľmi flexibilné pri usporadúvaní a zobrazovaní informácií. Základnými súčasťami, ktoré sú využívané pri tvorbe databázových objektov v databáze, sú tabuľky, vzťahy, obmedzenia a pohľady (Oppel, 2012).

Tabuľka je v relačných databázach hlavnou jednotkou na ukladanie dát. Každá tabuľka predstavuje entitu, každý riadok tabuľky predstavuje jeden výskyt tejto entity a každý stĺpec predstavuje atribút tejto entity.

Vzťah znázorňuje prepojenie medzi tabuľkami relačnej databázy. Je to prepojenie primárneho kľúča jednej tabuľky s cudzím kľúčom druhej tabuľky. Vďaka vzťahom je možné spojiť dáta z viacerých tabuliek a vybrať pritom len tie údaje, ktoré sú pre používateľa dôležité.

Obmedzenie je pravidlo, ktoré určuje, aké hodnoty je možné vkladať do databázového objektu. Po nastavení obmedzenia nie je možné uložiť do databázového objektu iné hodnoty ako sú povolené obmedzením. Zvyčajne sa aplikuje na tabuľku alebo atribút. Existuje niekoľko typov databázových obmedzení:

- Obmedzenie not null zamedzuje použitie hodnoty null. Hodnota null zodpovedá neznámej hodnote,
- obmedzenie primárneho kľúča zaručuje, že hodnoty stĺpca ktorý obsahuje toto obmedzenie, budú v rámci tabuľky jedinečné. Ak je toto obmedzenie použité na viacero stĺpcov tabuľky, tak kombinácia týchto stĺpcov vytvára zložený primárny kľúč. Hodnoty v stĺpci sa teda môžu opakovať, no kombinácia hodnôt v stĺpcoch tvoriacich primárny kľúč musí byť jedinečná. Obmedzenie primárneho kľúča sa väčšinou definuje pomocou indexu, čo je špeciálny typ databázového objektu slúžiaci na zrýchlenie vyhľadávania v stĺpcoch tabuľky databázy,
- obmedzenie jedinečnosti zaisťuje, že stĺpec alebo skupina stĺpcov tabuľky, pre ktoré je toto obmedzenie definované budú obsahovať jedinečné hodnoty.

V tabuľke je možné definovať viac obmedzení jedinečnosti. Takmer vždy sa toto obmedzenie implementuje pomocou indexu,

- referenčné obmedzenie dohliada, aby pre každý cudzí kľúč v tabuľke, existoval zodpovedajúci kľúč v nadradenej tabuľke,
- obmedzenie check kontroluje hodnotu stĺpca, pričom výsledkom je logická hodnota. Ak je výsledkom logická hodnota true, tak je možné zapísať hodnotu do tabuľky. Ak je výsledkom logická hodnota false, tak je zápis údaju do tabuľky zamietnutý (Oppel, 2012; Connolly, Begg, 2005).

Pohľad si je možné predstaviť ako virtuálne tabuľky zobrazujúce informácie, ktoré spĺňajú kritériá používateľov. Neslúži na ukladanie dát, ale iba na zobrazovanie dát z jednej alebo viacerých databázových tabuliek. Výhodou je možnosť skrývať stĺpce a tabuľky, ktoré používateľ nepotrebuje alebo nie je oprávnený zobrazovať (Drlík, Skalka, 2007).

Nerelačné databázy vynikajú v dátových kolekciami s veľkým objemom dát a tam, kde je očakávané časté spracúvanie dát alebo ich rýchle pribúdanie. Nerelačné databázy nevyužívajú relačný model. Komplexnosť údajov nie je uchovávaná v databázovej štruktúre ako pri relačných databázach, ale v štruktúre dát nazývanej dokument. Každý takýto dokument potom obsahuje všetky údaje patriace k danej entite (Membrey, Plugge, Hawkins, 2010).

1.4 DÁTOVÉ MODELÝ

Tvorbe samotnej databázy predchádza vytvorenie databázového návrhu. Cieľom návrhu databázy je vytvoriť model časti reálneho sveta. Tento súbor konceptuálnych prostriedkov používaný pre popis údajov, vzťahov medzi týmito údajmi a ich sémantiky, nazývame dátový model. Nezaobera sa operáciami, ktoré sa budú na dátach vykonávať, ale organizáciou dát. Dátové modely je možné na základe toho, s akými elementárnymi prvkami pracujú, rozdeliť na konceptuálny, logický a fyzický model (Merson, 2009).

Konceptuálne modely obsahujú informácie o entitách vzťahoch medzi nimi a tiež vlastnostiach problémovej domény. Umožňujú priamu definíciu obmedzení a sú flexibilné. Existuje viacero modelov patriacich do tejto skupiny. Tento typ modelu je najvhodnejším modelom pre komunikáciu so zúčastnenými stranami.

Logické modely sa vyvinuli z konceptuálnych modelov, ktoré boli založené na objektoch, na modely založené na záznamoch. Zaoberajú sa organizáciou dát v databáze

nezávisle od akéhokoľvek konkrétneho systému riadenia bázy dát, teda neobsahujú informáciu o spôsobe ukladania údajov do databázy. Najvyužívanejšími modelmi z tejto skupiny sú relačný model, hierarchický a sieťový model. Logické modely teda popisujú logickú štruktúru databázy. Každému atribútu je definovaný údajový typ, v ktorom je určený presný počet polí. Pole má väčšinou nemennú veľkosť (Drlík, Skalka, 2007).

Fyzické modely sú závislé na konkrétnom systéme riadenia bázy dát a ich cieľom je navrhnúť dátovú štruktúru prispôsobenú pre konkrétnu aplikáciu a databázový systém využívaný touto aplikáciou. Zahŕňajú optimalizáciu systému, napríklad spájanie a rozdeľovanie entít, duplikovanie dát a vytváranie indexov. Fyzické modely nie sú často používané (Merson, 2009).

1.5 ENTITNO-RELAČNÝ MODEL

Medzi najpoužívanejšie dátové modely patrí entitno-relačný model. Pri návrhu databázy je nutné model zostaviť tak, aby bol schopný zodpovedať akúkoľvek otázku týkajúcu sa údajov v databáze. Ďalším cieľom je obmedzenie a minimalizácia možných redundancií, teda zbytočného opakovania údajov v databáze (Drlík, Skalka, 2007).

Entity sú predmety z reálneho sveta o ktorých zhromažďujeme určité dáta. Nie všetky veci sú entity, iba tie veci o ktorých je potrebné ukladať dáta. Entita musí mať atribúty a vzťahy. Atribút je jednotka faktov charakterizujúca entitu. Mal by byť atomický, teda jeho hodnota by nemala byť deliteľná z pohľadu modelovanej domény.

Vzťahy sú prepojenia medzi reláciami, ktoré sú tvorené za účasti vzájomne si odpovedajúcich stĺpcov v prepojených reláciách. Na vytvorenie vzťahu medzi reláciami sa do záznamu relácie, s ktorou má byť vytvorené prepojenie, vloží cudzí kľúč zodpovedajúci primárnemu kľúču záznamu v druhej relácii. Relácie so vzťahom sa potom nazývajú účastníkmi vzťahu. Najčastejšie sa vyskytujúci je binárny vzťah, čiže vzťah s dvoma účastníkmi vzťahu, ale existujú aj unárne a ternárne vzťahy. Pri určovaní vzťahu medzi reláciami je možnosť výberu medzi povinne a voliteľne existujúcim vzťahom. Túto vlastnosť nazývame voliteľnosť, alebo aj parcialita. Určuje voliteľnosť alebo povinnosť existencie vzťahu, teda či sa relácia musí zúčastniť vzťahu (Oppel, 2012).

Reláciami sa v relačnom modeli nazývajú štruktúry pripomínajúce tabuľky, ktoré obsahujú atribúty entít. Relácia musí obsahovať primárny kľúč, aby bolo vždy možné jednoznačne identifikovať každý záznam. Reláciu v konkrétnom dátovom návrhu nazývame tabuľkou.

Kardinalita vzťahu určuje najväčší počet záznamov jednej relácie, ktoré možno asociovať s daným záznamom druhého účastníka vzťahu. Existujú tri typy kardinality, ktoré sa používajú pri návrhu databáz, a to:

- Vzťah 1:1 predstavuje vzťah kde je jeden záznam relácie asociovaný práve s jedným výskytom v inej relácii. V praxi sa tento vzťah využíva zriedkavo,
- vzťah 1:n sa používa, keď je jeden záznam relácie asociovaný s viacerými záznamami v relácii, s ktorou sú účastníkmi vzťahu. Vzťah 1:n je najčastejšie sa vyskytujúcim vzťahom. Primárna relácia je spravidla tá, ktorá sa nachádza na strane 1 a cudzia relácia sa nachádza na strane n,
- vzťah m:n sa aplikuje, ak každý záznam prvej relácie môže byť asociovaný s viacerými záznamami druhej relácie, a zároveň každý záznam druhej relácie môže byť asociovaný s viacerými záznamami prvej relácie. Tento vzťah nemôže byť modelovaný priamo, ale využíva spojovaciu reláciu, ktorá má s účastníkmi vzťahu vzťah 1:n (Gravelle, 2010).

Redundancia dát je nepotrebné opakovanie údajov v databáze, ktoré spôsobuje tri problémy. Prvým problémom je, že opakujúce sa dáta zbytočne zaberajú miesto. Druhý problém nastáva, keď je potrebné zmeniť hodnotu redundantného údaje. Vtedy je nutné zmeniť hodnotu v každom poli, v ktorom sa daný údaj vyskytuje. Tretí problém nastáva, keď sa pri menení dát zabudne na nejaký záznam, čoho dôsledkom je nekonzistentnosť dát (Drlik, Skalka, 2007).

Normalizácia databáz je dôležitá z hľadiska predchádzania problémov, ktoré sa môžu prejaviť pri aktualizovaní údajov relácií. Tieto problémy Edgar Codd nazval anomálie. Známe sú tri typy anomálií:

- Anomália pri vkladaní je situácia, keď vloženie nového riadka do tabuľky nie je možné uskutočniť kvôli umelej závislosti na inej relácii,
- anomália pri odstraňovaní nastáva, keď odstránenie údajov o jednej entite spôsobí nežiaducu stratu dát charakterizujúcich inú entitu,
- anomália pri aktualizácii je situácia, keď si aktualizácia jedinej údajovej hodnoty vyžaduje súčasne aktualizáciu viacerých riadkov alebo vektorov súradníc. Dôsledkom môže byť, že pri aktualizácii takýchto redundantných údajov sa na niektoré môže zabudnúť, čo spôsobuje nekonzistentnosť údajov (Oppel, 2006).

1.6 CASE NÁSTROJE

CASE nástroje sú určené na počítačom podporované systémové inžinierstvo. Ich prednosťou je zjednodušenie a automatizácia niektorých činností v procese návrhu systému. Medzi niektoré z výhod CASE nástrojov v oblasti návrhu databáz patria:

- Vytváranie dátového návrhu vo forme diagramu, ktorý je ľahšie zrozumiteľný a spracovateľný ako vo forme textu,
- zrýchlenie návrhu pomocou použitia preddefinovaných vzťahov. Pri tvorbe vzťahu medzi tabuľkami stačí zvoliť typ vzťahu a cudzie kľúče budú doplnené automaticky,
- generovanie SQL skriptu z vytvoreného diagramu. Vygenerovaný SQL skript potom obsahuje všetok kód potrebný k vytvoreniu databázy zodpovedajúcej návrhu,
- spätné vytvorenie dátových návrhov z existujúcich kódov,
- generovanie dokumentácie dátového návrhu (Singh, 2011; Coronel, Morris, 2014).

Medzi najpoužívanejšie CASE nástroje patria Enterprise Architect, Oracle SQL Developer Data Modeler a MySQL Workbench. Práve MySQL Workbench sme využili pri tvorbe metodiky, pretože je určený na tvorbu dátových návrhov a tiež je ľahko dostupný.

1.7 DÁTOVÁ INTEGRITA

Dátová integrita znamená presnosť, konzistentnosť a platnosť dát. Zaraďuje sa medzi najdôležitejšie hľadiská návrhu databáz, pretože jej nedodržanie spôsobuje nepresnosť alebo až neplatnosť dát. Dátová integrita je rozdelená na 4 typy:

- Doménová integrita je integrita na úrovni atribútu. Zaisťuje, že hodnoty v každom atribúte sú presné, konzistentné a platné. Hodnota atribútu v každom zázname musí zodpovedať požadovanému typu. Pre atribút je možné definovať dátový typ, dĺžku, veľkosť a rozsah hodnôt, či je možné uložiť hodnotu null a či hodnoty musia byť jedinečné. Dátové typy umožňujú obmedzovať aký typ dát sa dá vkladať do daného atribútu, napríklad dátový typ integer povoľuje ukladanie len celočíselných hodnôt,

- entitná integrita zabezpečuje, že v tabuľke je každý záznam jedinečný a že tabuľka má primárny kľúč, ktorý je pre každý záznam jedinečný a neobsahuje hodnotu null,
- referenčná integrita zabezpečuje, že dáta v dvoch tabuľkách, ktoré spolu súvisia, sú prepojené vzťahom. Tiež sa stará o to, že dáta v tabuľkách sú synchronizované kedykoľvek, keď dochádza k vkladaniu, aktualizovaniu alebo odstraňovaniu dát v tabuľke,
- business pravidlá nie sú založené na štruktúre databázy, ale na spôsobe, akým organizácia svoje dáta chápe a používa. Môžu ovplyvňovať obmedzenia atribútov, typ hodnôt atribútov, interval hodnôt a iné. Príkladom business pravidla je, že dátum zarezervovania pobytu musí byť menší ako dátum nástupu na pobyt (Hernandez, 2006).

1.8 METODIKA NÁVRHU RELAČNÝCH DATABÁZ

Pred začatím samotného návrhu databázy je nutné určiť požiadavky, ktoré má výsledný návrh spĺňať. Predíde sa tak nesprávnemu návrhu a jeho zmenám počas tvorby, ktoré spôsobujú jeho predĺženie. Tieto sú zapríčinené najmä nesprávnym alebo neúplným zadefinovaním riešenej úlohy.

1.8.1 Vytvorenie tabuliek

Návrh databázového modelu začína vytvorením tabuliek, ktoré sú základnými prvkami dátového modelu a slúžia na ukladanie dát o entitách. Tabuľky definujeme podľa toho, aké dáta má databáza ukladať. Aby bolo možné jednoducho rozlíšiť, aké dáta tabuľka obsahuje, je potrebné zvoliť jedinečný názov každej tabuľky. Pomenovanie tabuľky musí charakterizovať jej dáta, aby ktokoľvek podľa názvu vedel, čo obsahuje. Názov tabuľky by sa mal skladať z čo najmenšieho počtu slov a nemal by v sebe mať skratky. Každá tabuľka reprezentuje súbor objektov, preto je vhodné názov tabuľky uvádzať v množnom čísle (Hernandez, 2006).

1.8.2 Definovanie atribútov

Ďalším krokom návrhu je priradiť tabuľkám atribúty, ktoré danú tabuľku charakterizujú. Atribútmi sú všetky dáta, ktoré potrebujeme o entite uchovávať a danú entitu charakterizujú. Atribút reprezentuje jednu charakteristiku entity, čiže obsahuje len jednu hodnotu. Preto sa názov atribútu uvádza v jednotnom čísle. Hodnoty atribútov nesmú byť na sebe závislé. Napríklad vypočítané atribúty sú závislé na hodnotách iných

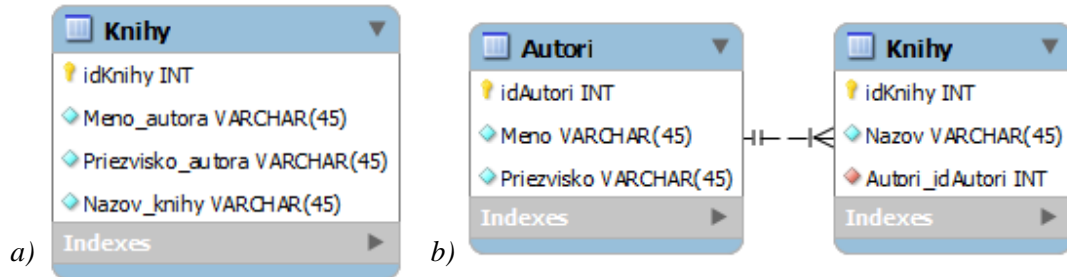
atribútov. Pri editovaní hodnoty, na ktorej je závislý vypočítaný atribút, nie je hodnota tohto atribútu automaticky aktualizovaná, ale musí ju aktualizovať používateľ pri každej zmene hodnoty.

Problémy so získavaním a triedením informácií nastávajú aj pri použití atribútov, ktorých hodnota je zložená z viacerých položiek. Situáciu je možné vyriešiť tak, že pre každú položku hodnoty takéhoto atribútu vytvoríme samostatný atribút. Príkladom atribútu, ktorého hodnota je zložená z viacerých položiek, je atribút adresa. Adresa sa bežne skladá z názvu ulice, mesta a poštového smerovacieho čísla. Ak sú tieto dáta uložené v jednom atribúte, tak je náročné napríklad triediť záznamy tabuľky podľa mesta, pretože je najprv potrebné získať názov mesta z atribútu adresa. Vytvorením samostatných atribútov pre ulicu, mesto a poštové smerovacie číslo sa tieto problémy vyriešia, keďže každý atribút bude obsahovať len jednu položku.

Nevhodné je aj použitie atribútov, ktoré môžu uchovávať viac ako jednu hodnotu. Takéto atribúty spôsobujú podobné problémy ako atribúty, ktorých hodnoty sú zložené z viacerých položiek. Vytvorenie samostatných atribútov pre každú z hodnôt v takomto prípade prináša ďalšie problémy. Pri zisťovaní, ktoré zo záznamov obsahujú danú hodnotu, je nutné prehľadávať všetky novovzniknuté atribúty. Ďalší problém nastáva, keď pri niektorom zo záznamov potrebujeme uchovávať viac hodnôt ako je na to určených atribútov. To by bolo možné dosiahnuť len vytvorením ďalšieho atribútu. Nevhodným riešením je aj vytvorenie záznamu pre každú hodnotu takéhoto atribútu, čiže vytvorenie viacerých záznamov, ktoré sa odlišujú len hodnotou jedného atribútu. Takýto postup by spôsobil problémy s redundanciou dát, pretože v tabuľke by existovalo viacero záznamov, ktoré by mali rovnaké hodnoty atribútov s výnimkou daného atribútu.

Príkladom môže byť tabuľka, v ktorej sa ukladajú informácie o autoroch a knihách ktoré napísali. Pre autorov, ktorí napísali viac ako jednu knihu by existovalo v tabuľke toľko záznamov, koľko kníh daný autor napísal. Ostatné atribúty tabuľky ako meno alebo priezvisko by však mali v týchto záznamoch rovnakú hodnotu. Vyriešenie problémov spojených s atribútmi s viacerými hodnotami spočíva v odstránení takýchto atribútov z tabuľky a vytvorení novej tabuľky, ktorá bude obsahovať odstránené atribúty a cudzí kľúč z pôvodnej tabuľky, ako je vidieť na Obr. 1 b). Cudzí kľúč obsahuje hodnotu primárneho kľúča z pôvodnej tabuľky, ktorý je v pôvodnej tabuľke unikátnym identifikátorom záznamu. V novej tabuľke je cudzí kľúč potrebný na

vytvorenie vzťahu so záznamami z pôvodnej tabuľky. V novej tabuľke sa cudzí kľúč môže opakovať vo viacerých záznamoch. Takéto redundantné dáta zabezpečujú vytvorenie vzťahu medzi dvoma tabuľkami. Tento typ redundantných dát je v databáze akceptovaný (Hernandez, 2006; Singh, 2011).



Obr. 1 a) Nesprávne definovanie atribútov, b) Správne definovanie atribútov

1.8.3 Určenie kľúčov

Každá tabuľka musí obsahovať aspoň jeden kandidátny kľúč, čiže atribút, ktorý jednoznačne identifikuje inštancie entity tabuľky a tak pomáha pri ochrane pred duplicitnými záznamami. Z kandidátnych kľúčov sa vyberie práve jeden, ktorý bude primárnym kľúčom. Kandidátny kľúč musí obsahovať unikátne hodnoty, ktoré nesmú obsahovať hodnotu null. Taktiež nie je vhodné za kandidátny kľúč zvoliť atribúty ako rodné číslo, keďže sa jedná o citlivé údaje. Kandidátny kľúč má mať nemennú hodnotu, ktorá by sa mala meniť len v nutných prípadoch. Kandidátny kľúč nemusí byť samotný atribút, ale môže byť tvorený kombináciou viacerých atribútov. Vtedy sa jedná o zložený kandidátny kľúč. V takomto prípade sa nevyžaduje jedinečnosť všetkých atribútov, ktoré zložený kandidátny kľúč tvoria, ale jedinečnosť kombinácie všetkých týchto atribútov.

Umelý kandidátny kľúč je atribút, ktorý sa vytvára len za účelom identifikácie záznamov a pôvodne sa v tabuľke nenachádzal. Umelý kandidátny kľúč obsahuje číselné hodnoty a funkcia auto-increment zabezpečí, že sa jeho hodnota pridaním nového záznamu zvýši. Zvolenie umelého kandidátneho kľúča za primárny kľúč je preferované pred ostatnými kandidátnymi kľúčmi. Príkladom výhody umelého primárneho kľúča je prípad, ak sa určí za primárny kľúč atribút priezvisko, tak nastáva problém, ak potrebujeme uchovávať dáta o viacerých osobách s rovnakým priezviskom. Priezvisko je v tomto prípade unikátnym identifikátorom, čiže musí obsahovať jedinečné hodnoty. Kvôli tomu nie je možné uchovávať záznamy o viacerých osobách, ktoré majú rovnaké priezvisko. Riešením je vytvorenie nového atribútu ID, ktorý bude

obsahovať jedinečné číselné hodnoty. Každý záznam tak bude možné identifikovať pomocou tohto čísla.

Pri voľbe primárneho kľúča sa vyberá taký kandidátny kľúč, ktorý je tvorený najmenším počtom atribútov, pretože spracúvanie je potom efektívnejšie. Primárny kľúč identifikuje záznamy tabuľky v rámci celej databázy, a preto je vhodné, aby jeho názov obsahoval názov tejto tabuľky. Okrem toho, že umožňuje jednoznačné identifikovanie záznamov v tabuľke, tiež napomáha pri predchádzaní problémov s integritou dát a redundanciou (Hernandez, 2006).

1.8.4 Definovanie obmedzení

Pomocou definovania obmedzení je možné zabezpečiť dátovú integritu, teda konzistentnosť, presnosť a platnosť dát. Databázové obmedzenia umožňujú určiť, aké dáta je možné v danom atribúte ukladať.

Obmedzenie typu not null sa aplikuje na povinné atribúty, čiže atribúty, ktoré musia mať vždy zadanú hodnotu. Obmedzenie typu not null musí byť vždy definované pre primárny kľúč a pri povinných vzťahoch aj pre cudzí kľúč.

Obmedzenie primárneho kľúča má za úlohu zaistiť, že atribút alebo atribúty, ktoré tvoria primárny kľúč, majú v každom zázname jedinečnú hodnotu a nesmú mať hodnotu null.

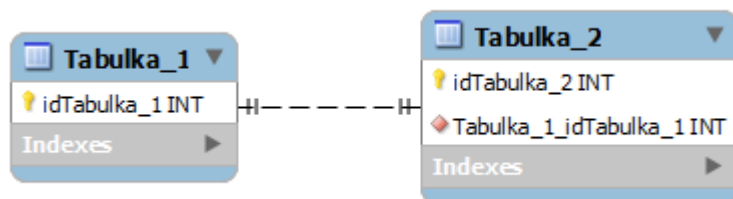
Referenčné obmedzenie označujúce cudzí kľúč zabezpečuje, že hodnota cudzieho kľúča v dcérskej tabuľke zodpovedá hodnote primárneho kľúča v niektorom zázname rodičovskej tabuľke. Tiež bráni odstráneniu záznamov z rodičovskej tabuľky, kvôli ktorému by hodnota cudzieho kľúča odkazovala na neexistujúcu hodnotu primárneho kľúča. Mnoho relačných databázových systémov umožňuje definovanie kaskádového odstránenia, ktoré zabezpečí, že pri odstránení záznamu z rodičovskej tabuľky sa odstránia aj záznamy dcérskej tabuľky, ktorých cudzí kľúč odkazuje na primárny kľúč práve zmazaného záznamu z rodičovskej tabuľky.

Dátový typ umožňuje určiť, aký typ hodnôt môže atribút obsahovať. Atribút s dátovým typom integer uchováva celočíselné hodnoty. Na ukladanie desatinných čísiel je v databázovom systéme MySQL možné použiť dátový typ decimal. Atribúty ako priezvisko alebo názov tovaru obsahujú reťazec, na čo sa používa dátový typ varchar. Dátový typ date umožňuje zápis dátumu. Tieto dátové typy patria medzi najpoužívanejšie, avšak pri tvorení návrhu je možné si vybrať vhodný typ zo širšej ponuky (Oppel, 2006).

1.8.5 Vytvorenie vzťahov medzi tabuľkami

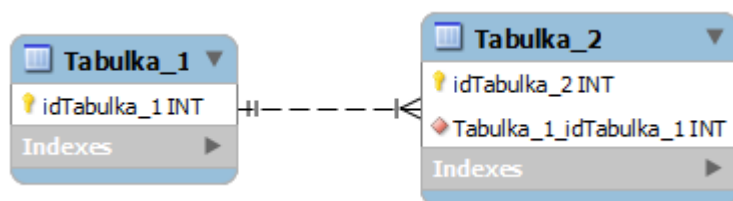
Po vytvorení tabuliek nasleduje v procese návrhu databázy vytvorenie vzťahov medzi týmito tabuľkami. Tabuľky je možné navzájom prepojiť použitím vzťahov s jednou z trojice kardinalít 1:1, 1:n a m:n.

Vzťah 1:1 sa používa, keď je potrebné prepojiť každý záznam prvej tabuľky s práve jedným záznamom z druhej tabuľky, a zároveň každý záznam druhej tabuľky s práve jedným záznamom z prvej tabuľky. Tento typ vzťahu sa používa len zriedkavo.



Obr. 2 Znáozornenie vzťahu 1:1

Vzťah 1:n definujeme pre dve tabuľky, kde každý záznam z prvej tabuľky môže byť prepojený s jedným alebo viacerými záznamami druhej tabuľky, a zároveň každý záznam druhej tabuľky môže byť prepojený s jediným záznamom z prvej tabuľky. Vzťah 1:n je najčastejšie používaným vzťahom, ktorý umožňuje eliminovať duplicitné dáta a znížiť tak redundanciu dát na najmenší možný počet. Graficky je tento vzťah zobrazený prerušovanou čiarou. Koniec tejto čiary sa rozvetvuje na strane tabuľky, z ktorej možno viacero záznamov prepojiť s jediným záznamom tabuľky na opačnom konci čiary.



Obr. 3 Znáozornenie vzťahu 1:n

Vzťah m:n sa používa na prepojenie tabuliek takým spôsobom, že jeden záznam z prvej tabuľky môže byť prepojený s jedným alebo viacerými záznamami z druhej tabuľky, a zároveň jeden záznam z druhej tabuľky môže byť prepojený s jedným alebo viacerými záznamami z prvej tabuľky. Tento typ vzťahu nie je možné v relačnom modeli realizovať priamo, ale vyžaduje vytvorenie pomocnej tabuľky.

Príkladom vzťahu m:n je vzťah medzi tabuľkou herci a tabuľkou filmy. V jednom filme môžu hrať viacerí herci a zároveň jeden herec môže hrať v jednom

alebo viacerých filmoch. Pri vzťahu $m:n$ je potrebné vytvoriť pomocnú tabuľku. Obsahuje cudzie kľúče oboch tabuliek zúčastňujúcich sa vzťahu. Tieto dva cudzie kľúče môžu tvoriť zložený primárny kľúč v prípade, že v pomocnej tabuľke nemôžu byť vytvorené viaceré záznamy, v ktorých by sa zhodovali oba cudzie kľúče. Jeden film môže byť prepojený s konkrétnym žánrom iba raz, pretože viacnásobné uvedenie toho istého žánru pri filme nemá význam. Napríklad k filmu sa neuvádza dvakrát alebo viackrát, že je komédia.

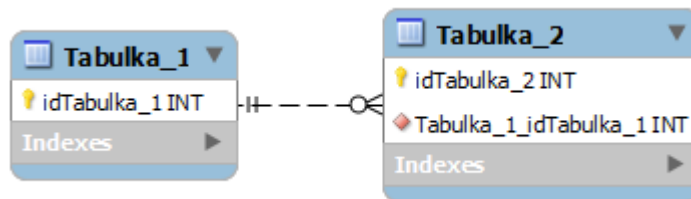
Naopak pri evidovaní skúšok, na ktoré je študent prihlásený, môže nastať situácia, že sa študent prihlási na opravný termín. V takomto prípade je v pomocnej tabuľke cudzí kľúč skúšky aj cudzí kľúč študenta zhodný s kľúčmi v zázname pre prvú skúšku. Aby bolo možné záznamy jednoznačne identifikovať, je potrebné vytvorenie umelého primárneho kľúča.



Obr. 4 Znáozornenie vzťahu $m:n$

Vo vzťahu je okrem kardinality dôležité určiť aj voliteľnosť účasti tabuľky na vzťahu. Ak je účasť na vzťahu povinná, tak každý záznam tabuľky musí byť prepojený aspoň s jedným záznamom druhej tabuľky vo vzťahu. Voliteľná účasť na vzťahu znamená, že ktorýkoľvek záznam tabuľky môže, no nemusí, byť prepojený so záznamom druhej tabuľky.

Príkladom voliteľnej účasti na vzťahu je prepojenie tabuliek Zákazníci a Objednávky. Ak na Obr. 4 predstavuje Tabulka_1 zákazníkov a Tabulka_2 objednávky, tak tabuľka Objednávky má voliteľnú účasť na vzťahu. Zákazník môže byť registrovaný aj bez toho, aby mal vytvorenú objednávku. Každý záznam tabuľky Zákazníci teda nemusí byť prepojený so žiadnym záznamom z tabuľky Objednávky, alebo môže byť prepojený s jedným alebo viacerými záznamami. Graficky sa voliteľnosť účasti na vzťahu znázorňuje symbolom prázdneho kruhu na strane tabuľky, ktorej účasť na vzťahu je voliteľná (Hernandez, 2006; Hopkins, 2010).



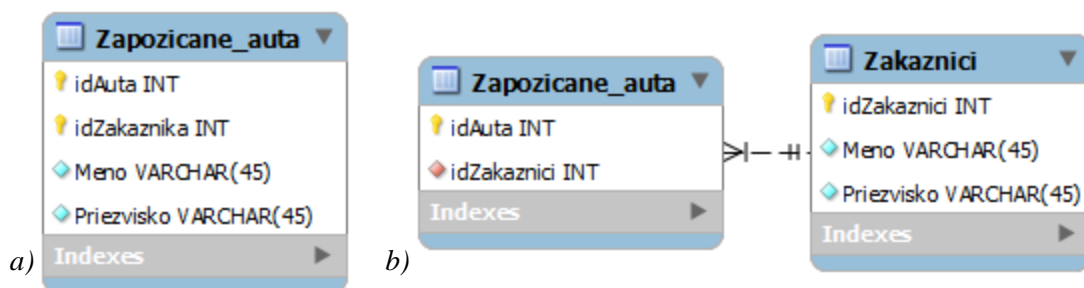
Obr. 5 Znáozornenie voliteľností účasti na vzťahu

1.8.6 Normalizácia návrhu

Normalizácia dátového návrhu je proces organizovania dát v databáze, ktorý zahŕňa elimináciu redundancie a nekonzistencie dát. Normalizovaná databáza nesmie spôsobovať anomálie pri vkladaní, aktualizácii a odstraňovaní dát. Počas normalizácie sa určuje, ktoré atribúty by mali byť zoskupené spolu v tabuľke. Definovaných je niekoľko normálnych foriem, pričom najdôležitejšie sú prvé tri, ktoré eliminujú anomálie spomínané v kapitole 1.5 Entitno-relačný model. Aby sa o databáze mohlo hovoriť, že je v konkrétnej normálnej forme, musí spĺňať pravidlá tejto formy a všetky predchádzajúce normálne formy (Sumathi, Esakkirajan, 2007).

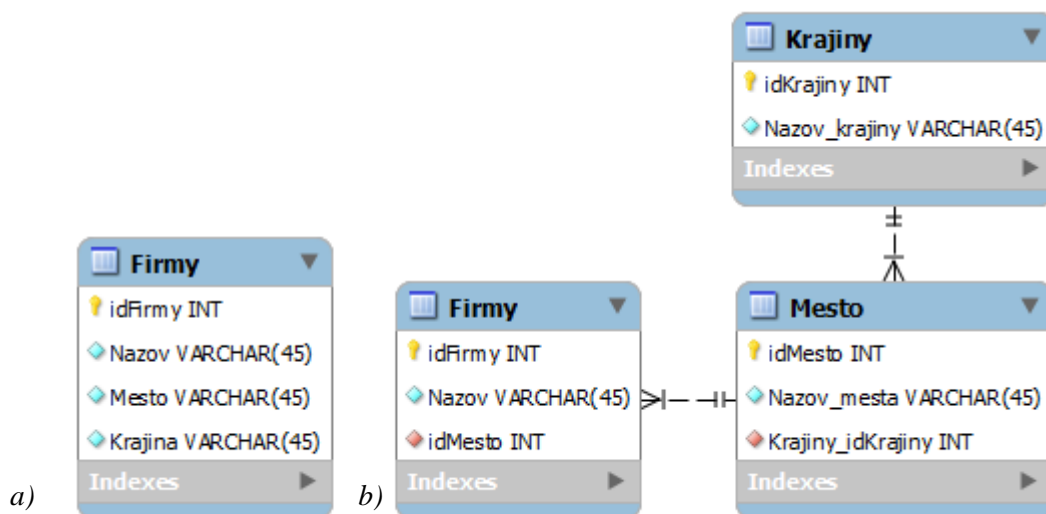
Tabuľka je v prvej normálnej forme práve vtedy, keď všetky atribúty uchovávajú práve jednu hodnotu. Žiadny atribút tabuľky teda nemôže v ktoromkoľvek zázname obsahovať viac ako jednu hodnotu. Vytvoriť tabuľku v prvej normálnej forme je vhodné už pri prvotnom definovaní atribútov, preto je dôležité riadiť sa zásadami tvorby atribútov, ktoré sú uvedené v kapitole 1.8.2 Definovanie atribútov.

Tabuľka je v druhej normálnej forme práve vtedy, keď je v prvej normálnej forme, a zároveň všetky atribúty, ktoré nie sú kľúčmi, sú úplne funkčne závislé na primárnom kľúči. Ak tabuľka obsahuje zložený primárny kľúč, tak neklúčové atribúty musia byť úplne funkčne závislé, čiže závislé na celom primárnom kľúči a nie na jednotlivých atribútoch, z ktorých sa primárny kľúč skladá. Príkladom je tabuľka Zapozicane_auta, ktorá obsahuje atribúty idAuta, idZakaznika, Meno a Priezvisko. Zložený primárny kľúč tvoria atribúty idAuta a idZakaznika. Atribúty Meno a Priezvisko sú funkčne závislé na atribúte idZakaznika a nie na celom kľúči, teda nie sú úplne funkčne závislé a tabuľka nie je v druhej normálnej forme. Túto situáciu je možné riešiť premiestnením atribútov, ktoré sú závislé, do novej tabuľky Zakaznici. Tabuľka Zapozicane_auta bude mať primárny kľúč stále zložený z atribútov idAuta a idZakaznika, avšak atribút idZakaznika bude cudzím kľúčom, ktorý vytvára vzťah s tabuľkou Zakaznici.



Obr. 6 Druhá normálna forma je: a) nesplnená, b) splnená

Tabuľka dosahuje tretiu normálnu formu, ak je v druhej normálnej forme, a zároveň všetky jej neklúčové atribúty sú navzájom nezávislé. Tabuľka Firmy, ktorá obsahuje atribúty idFirmy, Nazov, Mesto a Krajina, nie je v tretej normálnej forme, pretože atribút Krajina je funkčne závislý od atribútu Mesto. Riešenie takejto situácie spočíva vo vytvorení novej tabuľky Mesta, ktorá bude obsahovať atribúty ID a Nazov_mesta a vo vytvorení tabuľky Krajiny s atribútmi ID a Nazov_krajiny. Z tabuľky Firmy sa odstráni atribút Krajina a Mesto, atribút idMesto bude cudzím kľúčom. Taktiež v tabuľke Mesta bude cudzí kľúč idKrajina, ktorý odkazuje na záznamy v tabuľke Krajiny (Ricardo, 2011; Sumathi, Esakkirajan, 2007).



Obr. 7 Tretia normálna forma je: a) nesplnená, b) splnená

2 CIELE BAKALÁRSKEJ PRÁCE

Cieľom bakalárskej práce je vytvoriť materiál, ktorý poskytne čitateľom zaujímavým sa o problematiku modelovania relačných databáz, sadu riešených príkladov a pomôže im pochopiť princíp tvorenia relačných databáz.

Podciele:

- vyhľadať, vytriediť a analyzovať informačné zdroje, týkajúce sa danej témy,
- vytvoriť sadu riešených príkladov z oblasti návrhu relačných databáz,
- namodelovať relačné databázy tak, aby zodpovedali normálnym formám,
- spracovať postupy riešenia jednotlivých príkladov,
- prezentovať štandardné postupy návrhu,
- poukázať na časté problémy.

3 ZBIERKA PRÍKLADOV

Táto kapitola obsahuje zbierku príkladov a k nim vypracovaných dátových modelov. Tieto príklady majú stupňujúcu sa náročnosť, teda sa postupne zväčšuje aj počet tabuliek a zložitosť vzťahov medzi nimi. Príklady sú zamerané na situácie, ktoré nastávajú v reálnom svete. Medzi problémy, ktoré sa dajú riešiť prostredníctvom databáz, patria filmové databázy, evidencia knižnice, či databázy pre internetový obchod. Tieto ale aj ďalšie zahŕňajú nasledujúce podkapitoly, lebo sú vhodné na uvedenie do problematiky pre začínajúcich tvorcov databázových návrhov. Pri tvorbe príkladov bola pre inšpiráciu použitá stránka databaseanswers.org.

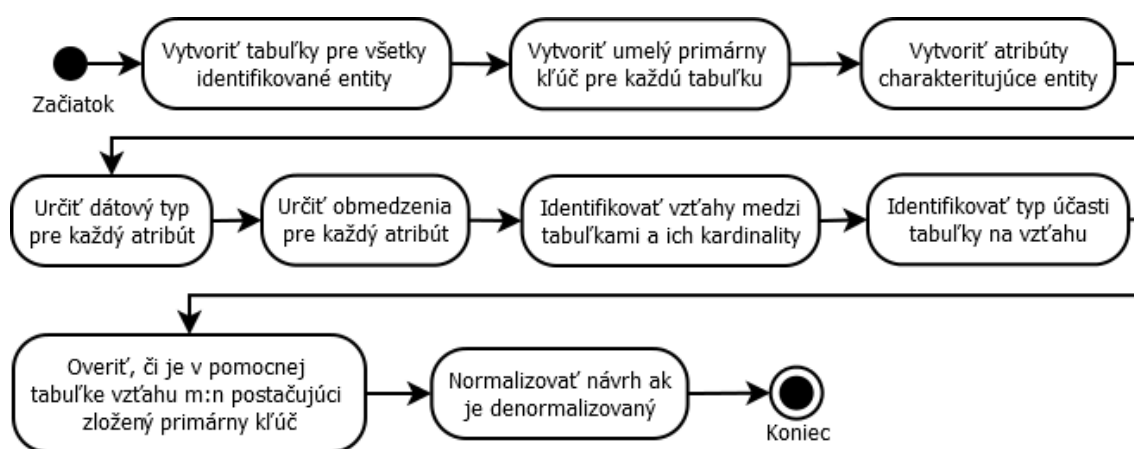
Prvý príklad, ktorým je filmová databáza, navyše obsahuje podrobný postup procesu návrhu databázy. Okrem tvorby tabuliek a ich atribútov, zahŕňa tiež určovanie obmedzení týchto atribútov. Venuje sa aj vytváraniu vzťahov medzi tabuľkami a určovaniu ich kardinality. Popisuje tiež potrebu vzniku pomocných tabuliek a normalizácie daného dátového návrhu.

Každý z príkladov má zadanie, ktoré špecifikuje požiadavky na výsledný návrh. Pre zložitejšie časti je uvedený spôsob riešenia vzniknutej situácie. Konečné riešenie príkladov je obsiahnuté prostredníctvom dátového návrhu. Niektoré z tabuliek nie sú v tretej normálnej forme. Príkladom môžu byť údaje o osobách, v ktorých je zahrnutá aj adresa. Tá by podľa normalizácie mala byť rozdelená na tabuľky ako sú Mesto a Krajina. Toto by však v reálnej databáze spôsobilo, že pre výber jednoduchého údaju ako je adresa, by bolo potrebné spájať dáta z niekoľkých tabuliek, čo je časovo náročnejšie.

3.1 FILMOVÁ DATABÁZA

Zadanie: Úlohou je vytvoriť dátový návrh pre filmovú databázu, ktorá bude slúžiť na uchovávanie dát o filmoch, hercoch, ktorí v nich hrajú a režiséroch. K filmu je okrem názvu a roku vydania potrebné uložiť aj žánre, do ktorých patrí.

Postup riešenia: Diagram aktivít zobrazený na Obr. 6 zachytáva postup tvorby návrhu relačnej databázy. Metodika uvádza, že sa môžu najskôr určiť atribúty a následne z nich vybrať primárne kľúče, ale tiež je spomenuté, že je vhodné používať umelé primárne kľúče. Nasledujúce riešenie zobrazuje práve tento prípad, teda využitie umelých kľúčov, ktoré je možné definovať skôr ako ostatné atribúty.



Obr. 8 Diagram aktivít procesu návrhu databázy

Prvým krokom je vytvorenie tabuliek pre všetky entity, ktoré vieme identifikovať ako potrebné. V tomto prípade sú to filmy, herci, režiséri a žánre. Následne sa do všetkých tabuliek vloží atribút ID a nastaví sa mu obmedzenie primárneho kľúča. Primárny kľúč musí mať aj obmedzenie not null, to však doplní MySQL Workbench automaticky po nastavení obmedzenia primárneho kľúča. V tabuľkách Režiseri a Herci sa vytvoria ďalšie atribúty, a to Meno a Priezvisko, ktoré budú mať obmedzenie not null a budú dátového typu varchar, keďže sa jedná o reťazec. Číslo v zátvorke určuje maximálny počet znakov, ktoré môže reťazec obsahovať. Keďže meno a priezvisko nebývajú príliš dlhé, tak v tomto príklade bude postačovať varchar(20), čiže reťazec, ktorý sa môže skladať z maximálne 20 znakov. Okrem týchto dvoch atribútov môžu tabuľky Režiseri a Herci obsahovať aj atribút Datum_narodenia, ktorý môže obsahovať aj hodnotu null, pretože v tomto prípade nie je nevyhnutné poznať dátum narodenia. Pre atribút, ktorý bude obsahovať dátumy, je určený dátový

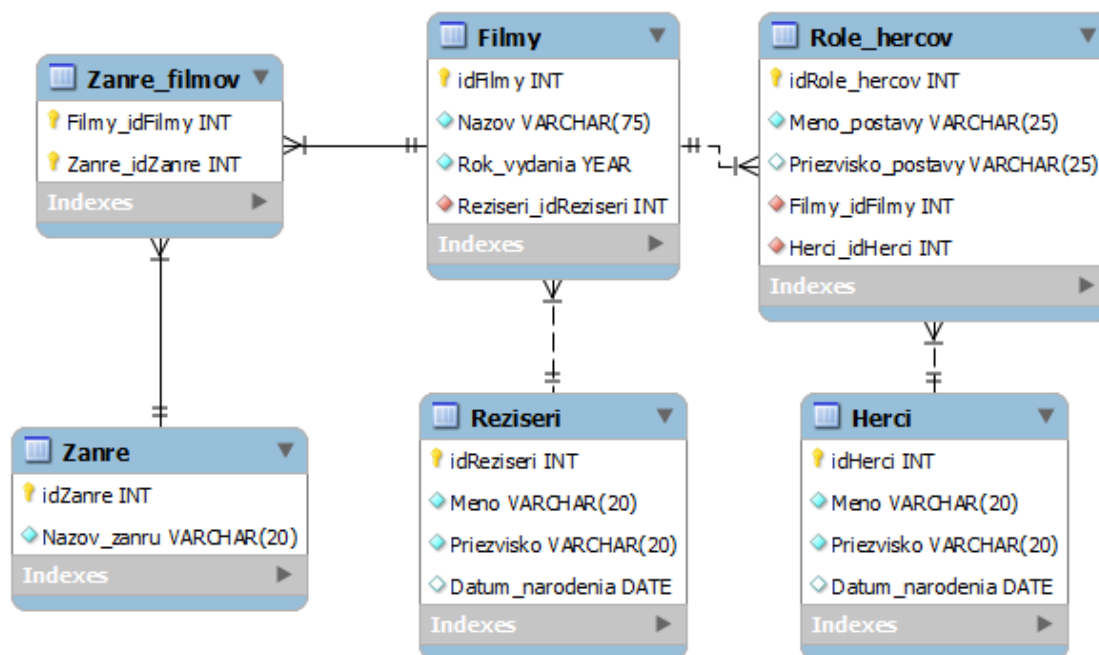
typ date. V tabuľke Zandre stačí okrem primárneho kľúča pridať atribút Nazov_zanru, ktorý je dátového typu varchar a jeho maximálna dĺžka je 20 znakov. Tabuľka Filmy bude uchovávať názvy filmov. Preto je potrebné v nej vytvoriť atribút Nazov. Keďže názov filmu sa môže skladať z viacerých slov, tak je vhodné určiť dĺžku reťazca tak, aby tento reťazec mohol obsahovať väčší počet znakov. Okrem názvu je potrebné o filme evidovať aj rok, v ktorom film vyšiel, na čo slúži atribút Rok_vydania. Tento atribút by mal byť dátového typu year alebo integer, ktoré sú pre toto použitie najvhodnejšie.

Po vytvorení všetkých tabuliek a atribútov, nasleduje v procese návrhu určenie vzťahov medzi jednotlivými tabuľkami. Keďže jeden režisér môže režírovať viacero filmov, a zároveň jeden film môže mať len jedného režiséra, použijeme vzťah s kardinalitou 1:n. Ten je zabezpečený cudzím kľúčom Reziseri_idReziseri v tabuľke Filmy, pričom tento cudzí kľúč odkazuje na primárny kľúč tabuľky Reziseri. Po vybratí vzťahu 1:n a zvolení tabuliek, ktoré sa na vzťahu zúčastňujú, nástroj MySQL Workbench automaticky vytvorí cudzí kľúč v tabuľke, ktorá bola zvolená ako prvá.

Pri vytváraní vzťahu medzi tabuľkami Filmy a Zandre je potrebná pomocná tabuľka, pretože jeden film môže patriť do viacerých žánrov a jeden žáner sa priradzuje viacerým filmom. Vzťah s kardinalitou m:n teda nie je možné modelovať priamo. Vzniknutá pomocná tabuľka Zandre_filmov má zložený primárny kľúč, ktorý je tvorený cudzími kľúčmi odkazujúcimi na tabuľky Filmy a Zandre. Tak ako cudzí kľúč, tak aj pomocnú tabuľku pri vzťahu m:n vytvorí MySQL Workbench automaticky. Analogicky postupujeme aj pri tvorení vzťahu medzi tabuľkami Filmy a Herci. Keďže v jednom filme hrá viac hercov a jeden herec môže hrať vo viacerých filmoch, tiež sa jedná o vzťah m:n. Vzniknutá pomocná tabuľka môže byť pomenovaná ako Role_hercov. Nakoľko však jeden herec môže mať v jednom filme niekoľko rolí, znamená to, že niekoľko záznamov môže mať rovnakú kombináciu cudzích kľúčov Filmy_idFilmy a Herci_idHerci. Primárny kľúč zložený z týchto cudzích kľúčov by preto nebol vhodný. Namiesto toho sa vytvorí umelý primárny kľúč idRole_hercov. Aby bolo možné evidovať aj meno postavy vytvorila sa atribúty Meno_postavy a Priezvisko_postavy, ktoré sú dátového typu varchar. Atribút Priezvisko_postavy môže obsahovať hodnotu null, pretože niektoré postavy sú známe len pod prezývkou. Zvyšné atribúty tejto tabuľky majú obmedzenie not null.

Vytvorený dátový návrh je v tretej normálnej forme, pretože žiadny z atribútov neobsahuje viacero hodnôt a ani hodnoty, ktoré sa skladajú z viacerých položiek, v každej tabuľke sú všetky atribúty úplne funkčne závislé na primárnom kľúči a všetky nekľúčové atribúty sú navzájom nezávislé.

Riešenie:



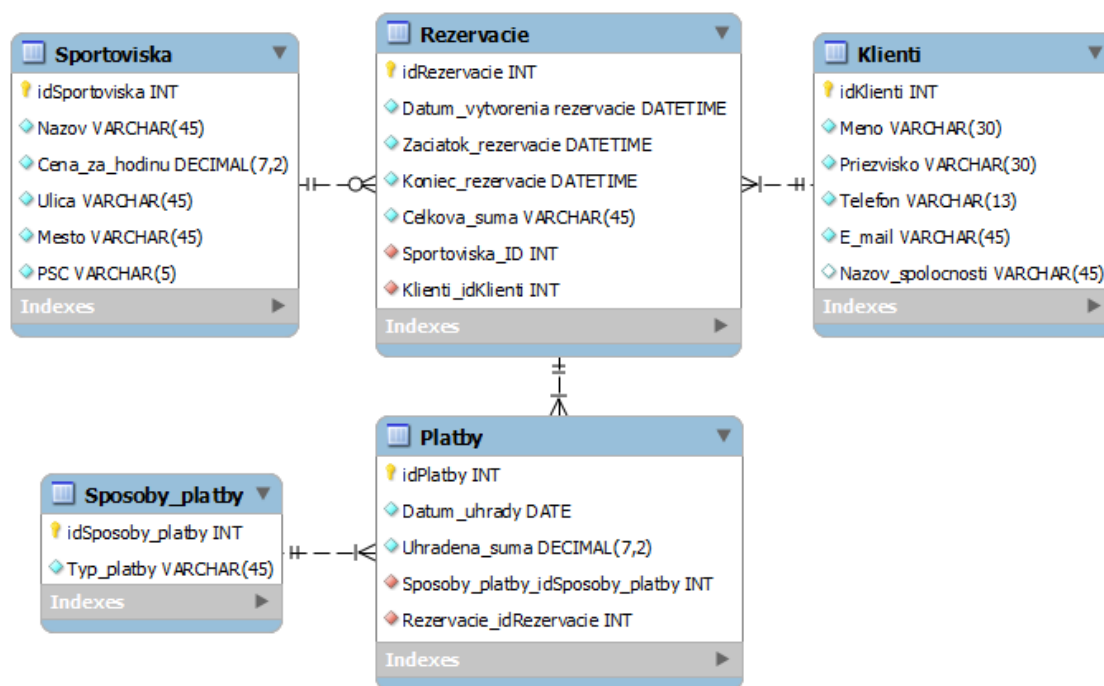
Obr. 9 Dátový návrh filmovej databázy

3.2 REZERVÁCIE ŠPORTOVÍSK

Zadanie: Dátový návrh pre rezerváciu športovísk má evidovať klientov, ktorí si rezervujú športovisko a dáta charakterizujúce dané športovisko. Klient poskytuje kontaktné údaje ako sú meno, priezvisko, či telefónne číslo a v prípade, že sa jedná o spoločnosť zadáva aj jej názov. Športovisko, ktoré si vyberá, má svoj názov, cenu za prenájom a adresu. Pre vytvorenie rezervácie je potrebné uviesť čas, v ktorom bude športovisko rezervované, ale aj celkovú sumu a spôsob jej uhradenia. Platba nemusí byť uskutočnená naraz, ale je možné najprv vyplatiť len zálohu a zvyšok doplatiť neskôr. Vhodné je pri rezervácii zaznamenávať aj dátum, kedy bola vytvorená.

Pomôcka: Telefónne číslo je dátového typu varchar. Tento typ je vhodný, lebo hoci sa jedná o číselný údaj, pracujeme s ním ako s reťazcom znakov a nevykonávame s ním žiadne matematické operácie.

Riešenie:



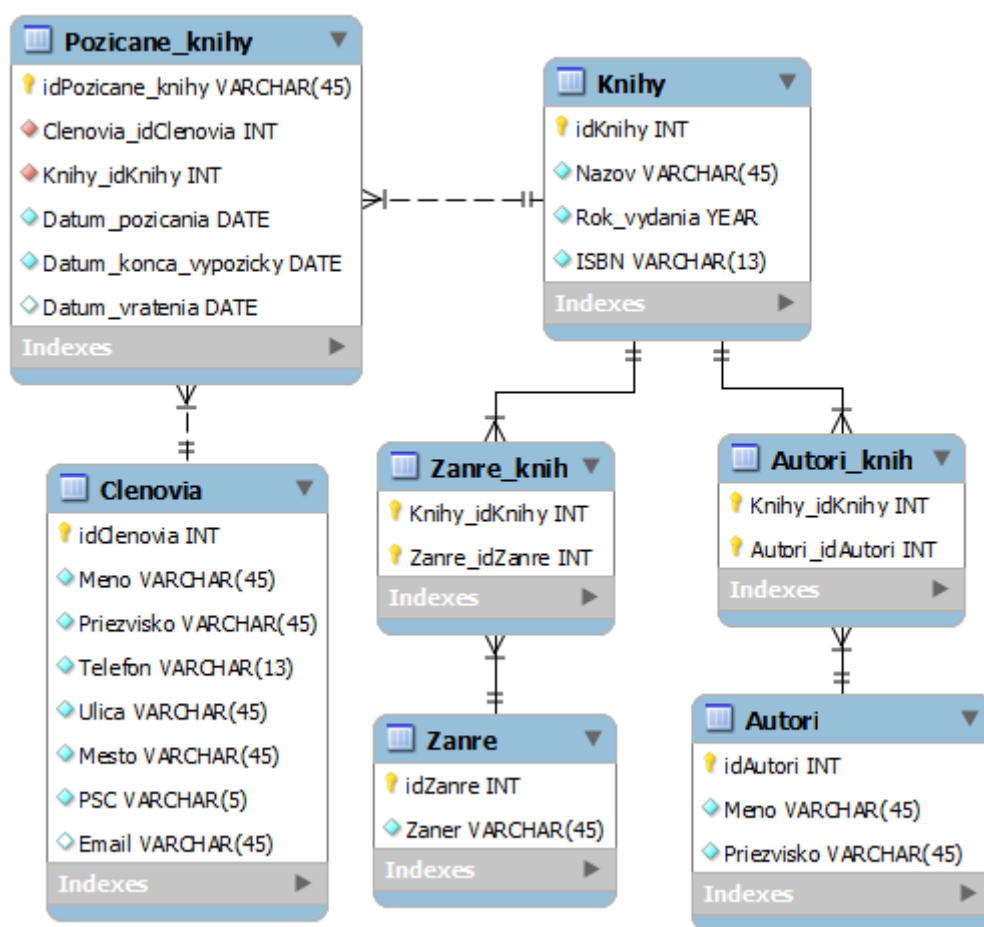
Obr. 10 Dátový návrh rezervácií športovísk

3.3 EVIDENCIA KNIŽNICE

Zadanie: Evidencia knižnice slúži na zaznamenávanie jej členov, dostupných kníh a ich výpožičiek. Dôležité sú kontaktné dáta členov a dáta charakterizujúce knihy, medzi ktoré patria jej názov, rok vydania a ISBN. Okrem toho databáza uchováva aj žáner, do ktorého sa kniha zaraďuje a tiež jej autora. V neposlednom rade je potrebné zapisovať dátum zapožičania, dátum ukončenia výpožičky knihy, ale aj dátum vrátenia knihy.

Pomôcka: Dátum vrátenia nie je pri vytváraní výpožičky známy, preto nadobúda hodnotu null. Pri určovaní vzťahov medzi tabuľkami Knihy a Zánre je potrebné si uvedomiť, že nielen jeden žáner charakterizuje viacero kníh, ale i jedna kniha sa môže radiť medzi viaceré žánre. Analogicky je určený vzťah aj medzi tabuľkami Autori a Knihy.

Riešenie:



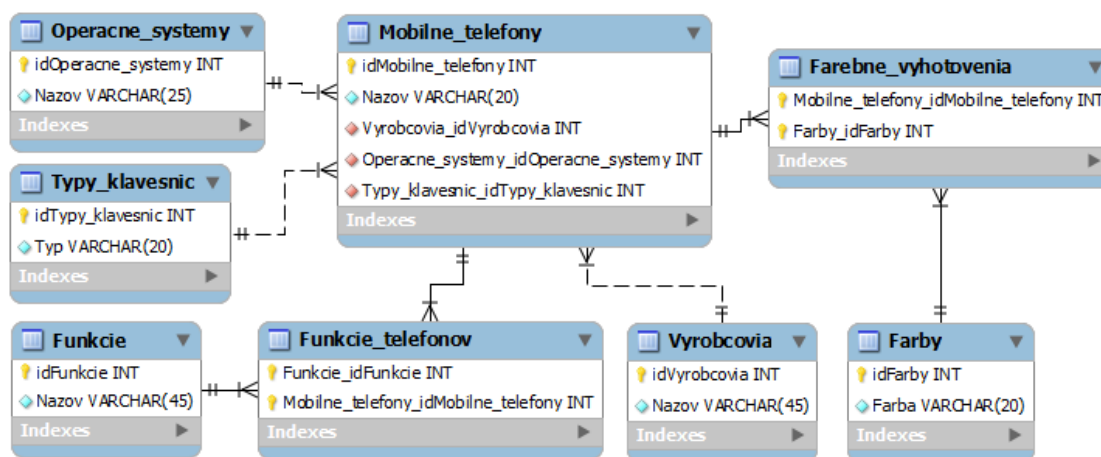
Obr. 11 Dátový návrh knižnice

3.4 DATABÁZA MOBILNÝCH TELEFÓNOV

Zadanie: Úlohou je vytvoriť návrh databázy mobilných telefónov. Každý telefón má uvedené pomenovanie modelu a názov výrobcu. Dôležitým údajom je aj operačný systém, ktorý telefón používa. Telefóny môžu mať rôzne typy klávesníc, ako sú dotyková, hardvérová alebo hardvérová QWERTY. Pre používateľov sú smerodajnými údaje o funkciách jednotlivých zariadení, ako napríklad WiFi či GPS navigácia. Výrobcovia často ponúkajú na výber viacero farebných kombinácií, aby zaujali širšie spektrum zákazníkov.

Pomôcka: Medzi farbami a mobilnými telefónmi existuje vzťah m:n, lebo model môže byť vyrábaný v rôznych farbách a každá z farieb môže byť použitá na viacerých telefónoch.

Riešenie:



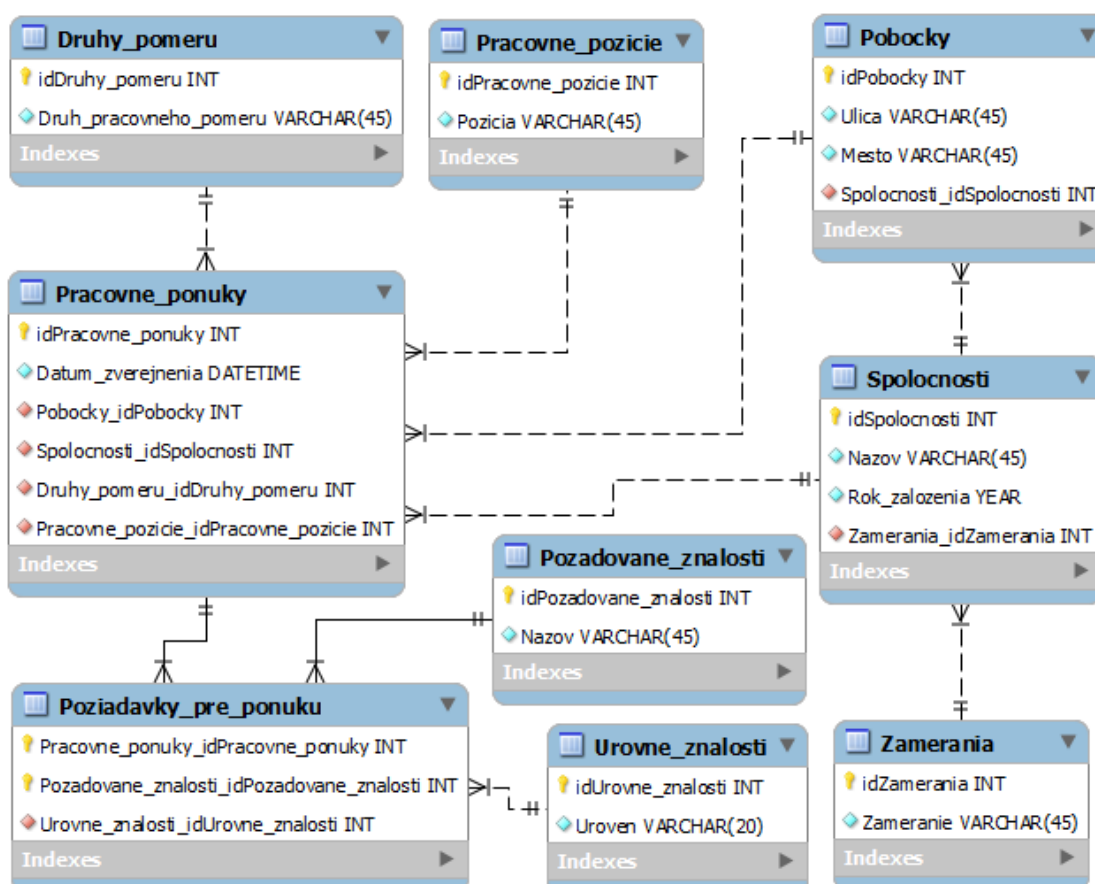
Obr. 12 Dátový návrh databázy mobilných telefónov

3.5 DATABÁZA PRACOVNÝCH PONÚK

Zadanie: Dátový návrh má evidovať pracovné ponuky predkladané rôznymi spoločnosťami. O ponuke sa zaznamenáva dátum jej zverejnenia a druh pracovného pomeru, teda či je to práca na plný úväzok, živnosť alebo na dohodu. Dôležitá je aj pozícia, na ktorú spoločnosť hľadá zamestnanca. Samotná spoločnosť musí uviesť svoj názov, rok založenia, zameranie svojej činnosti a adresy pobočiek, na ktoré sa ponuky vzťahujú. Zaznamenávajú sa i znalosti, ktoré spoločnosť od uchádzača požaduje, a ich úroveň.

Pomôcka: Medzi tabuľkami Pracovne_ponuky a Pozadovane_znalosti je vzťah m:n. Vzniká pomocná tabuľka Poziadavky_pre_ponuku, ktorá okrem zloženého primárneho kľúča obsahuje aj cudzí kľúč odkazujúci na záznamy tabuľky Urovne_znalosti. Tým sa zabezpečí určenie úrovne požadovanej znalosti.

Riešenie:



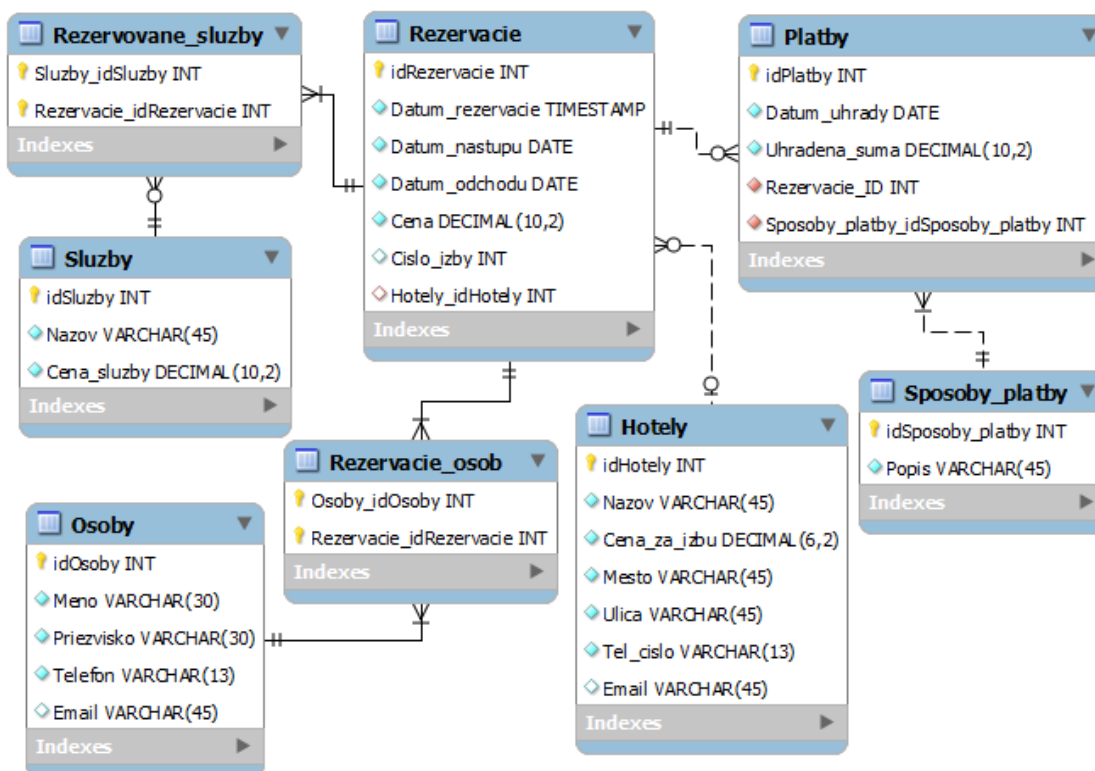
Obr. 13 Dátový návrh databázy pracovných ponúk

3.6 REZERVÁCIE KÚPEĽOV, ICH SLUŽIEB A HOTELOV

Zadanie: Úlohou databázy je ukladať údaje o rezerváciách kúpeľov, s možnosťou ubytovania sa v niektorom z ponúkaných hotelov. Rezervácia eviduje dátum nástupu a odchodu hostí, cenu za poskytnuté služby, v prípade ubytovania aj názov hotela a číslo izby. Uvádza sa aj spôsob platby vrátane konkrétnej sumy a dátumu úhrady. Osoba poskytuje o sebe kontaktné údaje. Databáza má obsahovať informácie o hoteloch, ktoré kúpele odporúčajú a sprostredkovávajú pobyt v nich. Okrem základných údajov, ako sú názov a adresa, zahŕňajú i cenu izby v danom hoteli.

Pomôcka: Vzhľadom na to, že kúpele je možné rezervovať na jeden deň, teda bez ubytovania, nemusí byť pri rezervácii uvedený hotel. Účasť hotela na vzťahu s rezerváciou je preto voliteľná. Taktiež nový hotel pridaný do databázy nemusí mať vytvorenú rezerváciu, preto účasť rezervácie na tomto vzťahu je tiež voliteľná. Vo vzťahu medzi rezerváciou a platbou je účasť platby na vzťahu voliteľná, lebo zálohu nie je nutné vyplatiť hneď pri zadávaní rezervácie.

Riešenie:



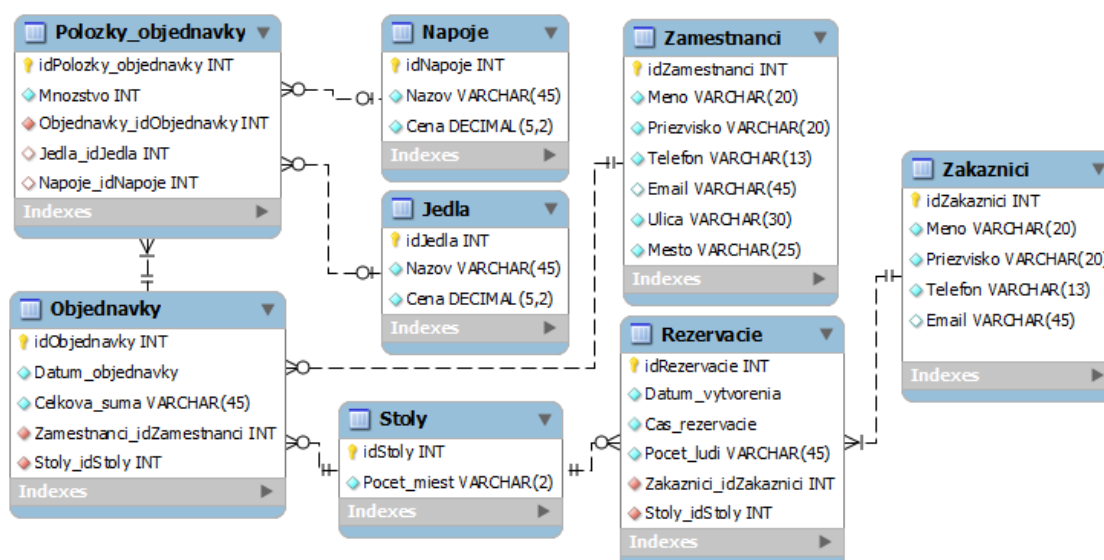
Obr. 14 Dátový návrh rezervácií služieb kúpeľov a hotelových izieb

3.7 REŠTAURÁCIA

Zadanie: Úlohou je vytvoriť návrh databázy, ktorá eviduje rezervácie stolov v reštaurácii a objednávky jedál. Zákazníci pri rezervácii uvádzajú svoje kontaktné údaje. Pri rezervovaní je potrebné uviesť počet ľudí, dátum a čas, na ktorý je rezervácia vytvorená a vybraný stôl. Pri zadávaní objednávky sa uvádza celková suma a tiež zamestnanec, ktorý ju vykonal. Objednávka pozostáva z nápojov a jedál. Reštaurácia uchováva kontaktné údaje o všetkých svojich zamestnancoch.

Pomôcka: Každý stôl nemusí mať rezerváciu ani objednávku, preto je ich účasť na vzťahu voliteľná. Podobne ani zamestnanec zatiaľ nemusel vytvoriť žiadnu objednávku. Objednávka sa môže skladať len z jedál alebo len z nápojov, preto je účasť tabuliek na vzťahu s tabuľkou Polozky_objednavky voliteľná. Keďže tabuľka Polozky_objednavky obsahuje aj atribút Mnozstvo, ktorý udáva počet rovnakých položiek na objednávke, tak nemôžu byť súčasne v zázname uvedené aj nápoje aj jedlá. Cudzí kľúč odkazujúci na záznam jednej z týchto tabuliek musí mať vždy hodnotu null. Tabuľky Jedla, Napoje a Polozky_objednavky teda nespĺňajú tretiu normálnu formu, pretože hodnota jedného z cudzích kľúčov je závislá na hodnote druhého primárneho kľúča.

Riešenie:



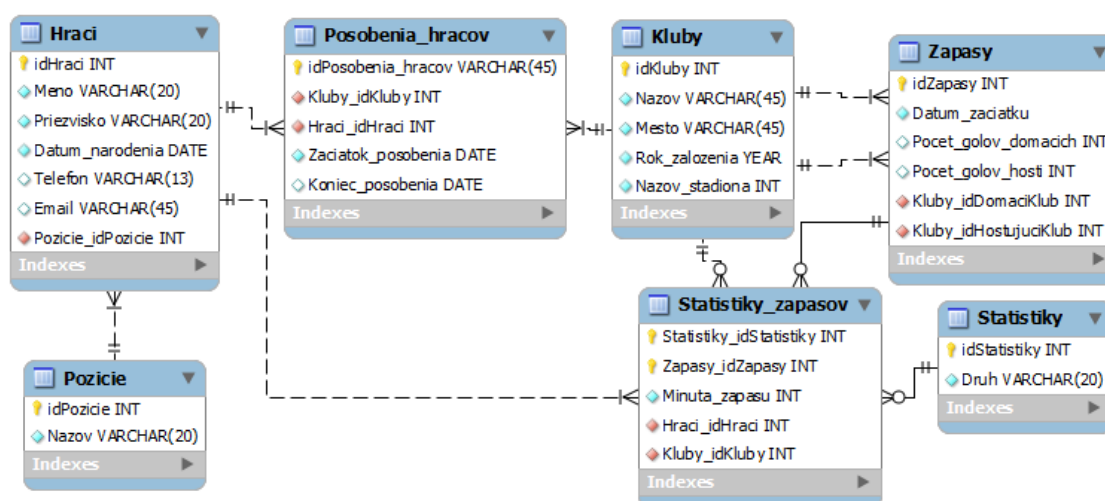
Obr. 15 Dátový návrh reštaurácie

3.8 HOKEJOVÁ LIGA

Zadanie: Úlohou je vytvoriť návrh databázy, ktorá eviduje priebeh hokejovej ligy. Údaje o kluboch zahŕňajú ich názov, mesto pôsobenia, rok založenia, názov štadióna a hráčov, ktorí v týchto kluboch hrajú alebo hrali v minulosti. Hráčov charakterizuje meno, priezvisko, dátum narodenia a prípadne iné kontaktné údaje. Dôležitá je aj pozícia, na ktorej hráči hrajú a obdobie pôsobenia v konkrétnom tíme. O zápasoch, ktoré počas sezóny prebiehajú, je nutné ukladať počet gólov, dátum zahájenia a miesto konania stretnutia. Jednotlivé štatistiky zápasov, ako sú góly či vylúčenia, sa zaznamenávajú spolu s časom, kedy k nim došlo a hráčom, ktorý sa o štatistiku zaslúžil.

Pomôcka: Na každý zápas proti sebe nastupujú dve mužstvá, z ktorých jeden je domáci a druhý hosťujúci klub. Kvôli tomu vznikajú medzi tabuľkami Zapasy a Kluby dva vzťahy. Jeden cudzí kľúč odkazuje na domáci a druhý na hosťujúci klub. Štatistika sa o zápase vedie až po stretnutí, preto je účasť tabuľky Statistika_zapasov na vzťahu s tabuľkou Zapasy voliteľná. Podobne je účasť tejto tabuľky voliteľná na vzťahu s tabuľkou Kluby, pretože kluby ktoré ešte neodohrali ani jeden zápas, nemajú žiadnu štatistiku. Keďže do databázy sa pridávajú dáta aj o nadchádzajúcich stretnutiach, tak nepoznáme výsledky takýchto zápasov a atribúty Pocet_golov_domacich a Pocet_golov_hosti môžu obsahovať hodnotu null.

Riešenie:



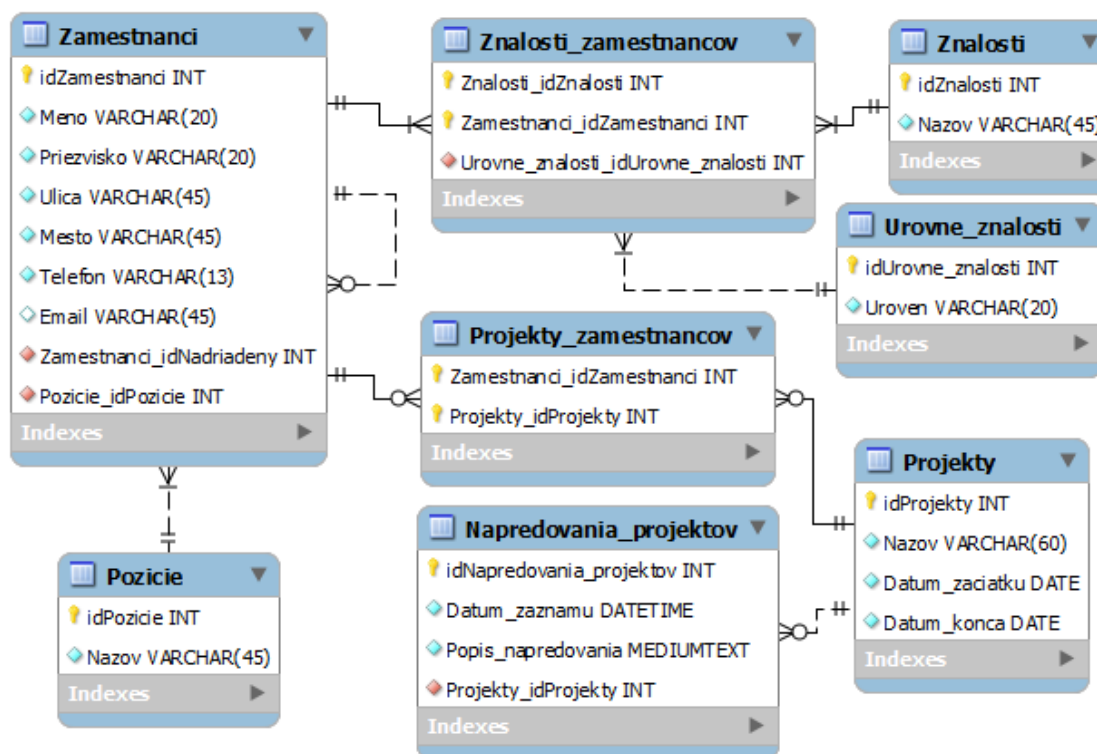
Obr. 16 Dátový návrh pre evidenciu hokejovej ligy

3.9 PRÁCA NA PROJEKTOCH

Zadanie: Cieľom je vytvoriť databázu zamestnancov pracujúcich na projektoch zadávaných zamestnávateľom. O zamestnancoch sa ukladajú kontaktné údaje a v prípade že majú nadriadeného, tak aj jeho údaje. Dôležité je tiež poznať pozície, ktoré pracovníci zastávajú, a tiež ich znalosti, ako aj úroveň týchto znalostí. Projekty majú názov a tiež dátum začiatku a ukončenia. Na každom z projektov môže pracovať niekoľko zamestnancov a ich napredovanie na projekte sa zaznamenáva krátkym popisom.

Pomôcka: Nadriadení zamestnancov sú evidovaní pomocou rekurzívneho vzťahu v tabuľke Zamestnanci, lebo sú tiež pracovníkmi daného zamestnávateľa, čiže majú záznam v tejto tabuľke. Tento vzťah je voliteľný, pretože každý zamestnanec nemusí mať nadriadeného. Pri zadávaní projektu ešte nie je určené, kto na ňom bude pracovať a podobne ani nový zamestnanec nemusí mať pridelený projekt.

Riešenie:



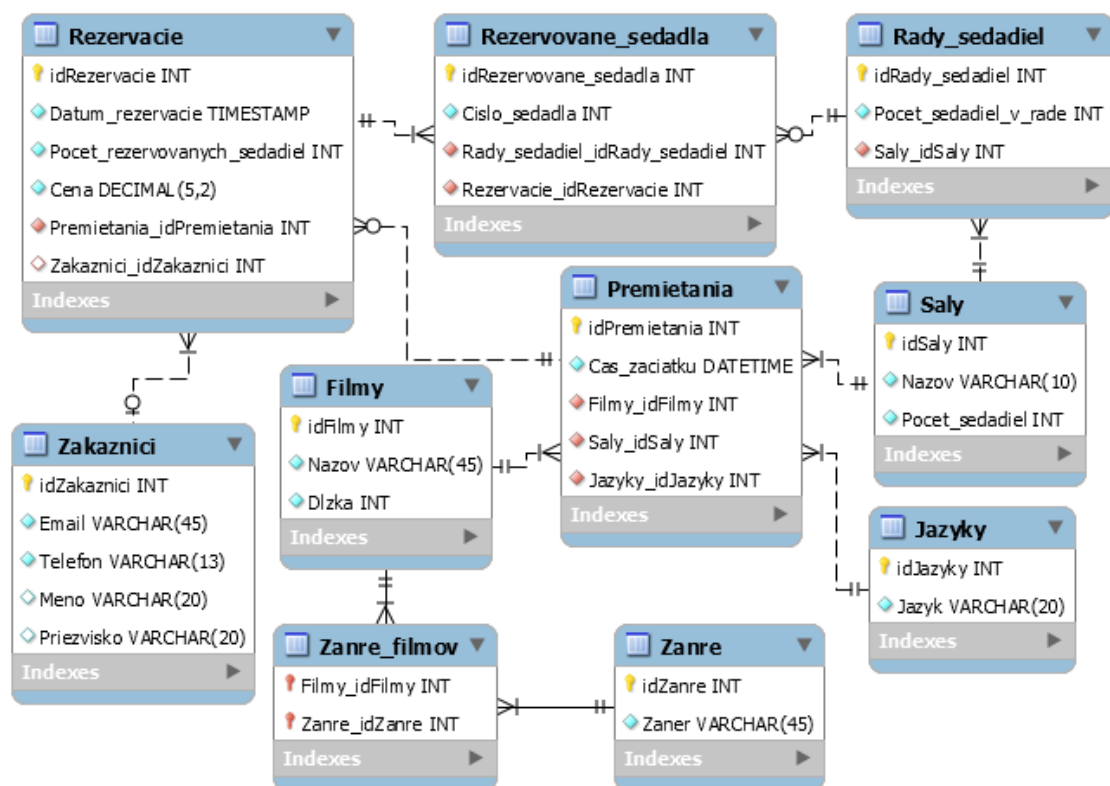
Obr. 17 Dátový návrh práce na projektoch

3.10 PREDAJ VSTUPENIEK DO KINA

Zadanie: Databáza pre kino slúži na uchovávanie dát o rezervácii miest na jednotlivé premietania. Pri rezervácii sa zadáva počet sedadiel, ich umiestnenie a cena za vstupenky. Zákazník si vyberá film a čas jeho premietania. O filme sa eviduje jeho dĺžka a žánre. Nakoľko kino obsahuje niekoľko kinosál, uvádza sa ku každému premietaniu aj názov sály, v ktorej sa uskutoční a jazyk tohto premietania. Rady sedadiel v týchto sálach môžu mať rôzny počet sedadiel. V prípade, že si zákazník rezervuje vstupenky cez internet, je potrebné aby zadal svoje kontaktné údaje ako email a telefón a ak sa chce zaregistrovať tak aj meno a priezvisko.

Pomôcka: Pri kúpe vstupeniek priamo v kine nežadáva zákazník svoje kontaktné údaje, preto je účasť záznamov tabuľky Zakaznici na vzťahu voliteľná. Nové premietanie filmu je možné do databázy pridať aj bez toho aby preň bola vytvorená rezervácia, takže účasť rezervácie na vzťahu s premietaním je voliteľná.

Riešenie:



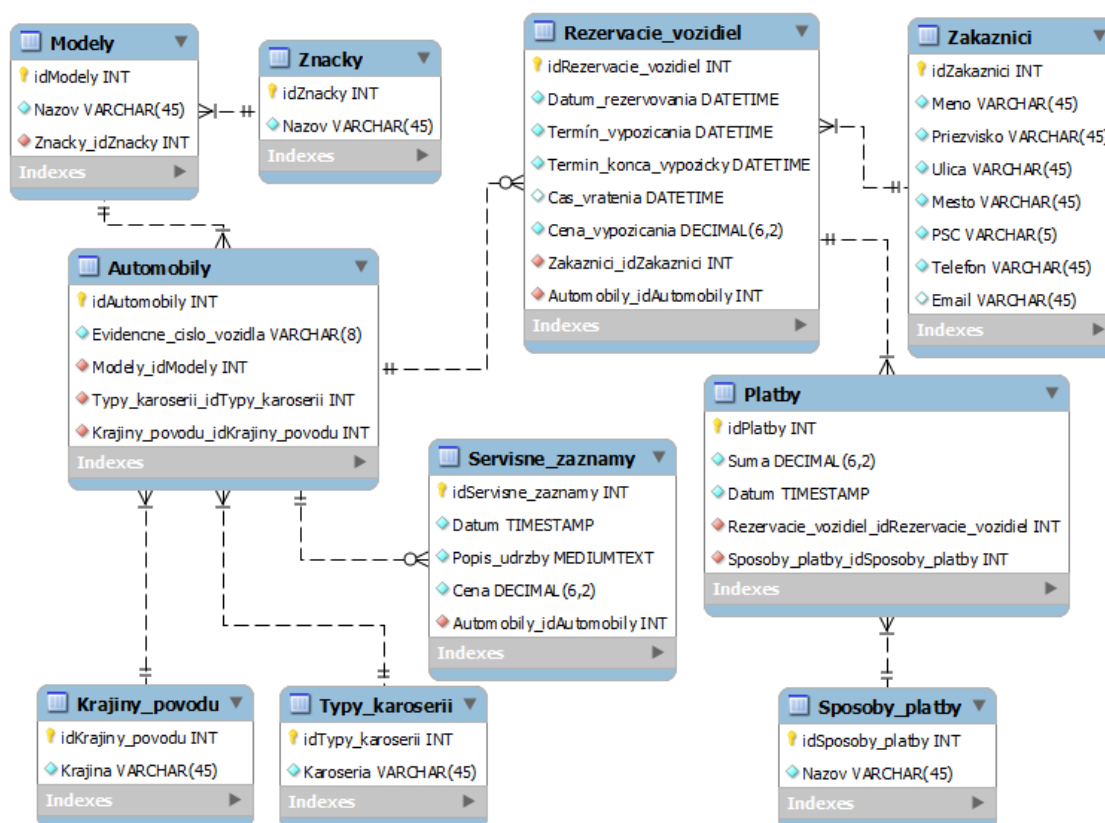
Obr. 18 Dátový návrh predaju vstupeniek do kina

3.11 POŽIČOVŇA AUTOMOBILOV

Zadanie: Úlohou je vytvoriť dátový návrh pre požičovňu automobilov. Zákazník poskytuje základné kontaktné údaje, ktoré sú potrebné pre uskutočnenie rezervácie. Samotná rezervácia obsahuje dátum a čas jej vytvorenia, termín vypožičania auta, konca výpožičky a skutočný čas, kedy zákazník vozidlo vrátil. Uvádza sa aj platba a spôsob jej uhradenia. Zákazník si zvyčajne automobil vyberá podľa modelu alebo typu karosérie, preto je tieto údaje potrebné v databáze zachytiť. Pre náročnejších zákazníkov, ale aj pre danú požičovňu, môže byť zaujímavá informácia o krajine pôvodu automobilu a servisné záznamy obsahujúce popis údržby a jej cenu.

Pomôcka: Rezervácia je s automobilom vo voliteľnom vzťahu, lebo pre automobil nachádzajúci sa v požičovni nemusí existovať rezervácia. Skutočný čas vrátenia vozidla nie je pri vytváraní rezervácie známy a dopĺňa sa až neskôr, preto môže tento atribút obsahovať hodnotu null.

Riešenie:



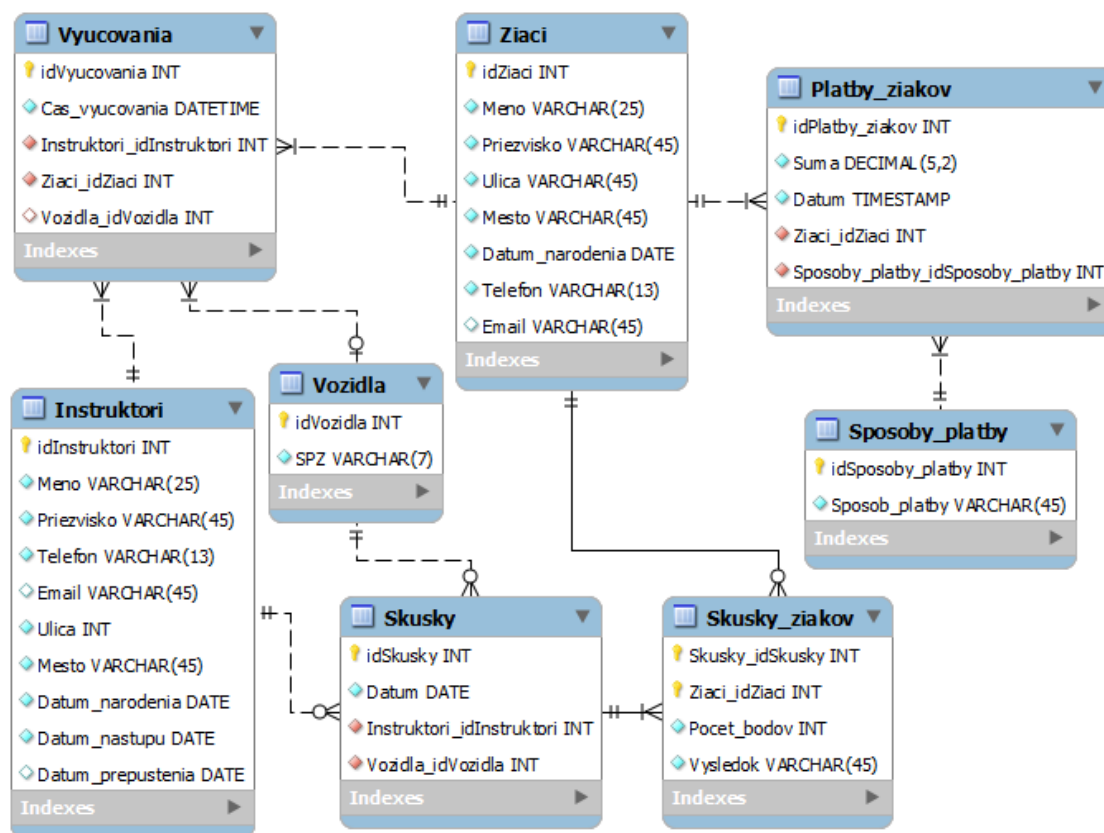
Obr. 19 Dátový návrh požičovne automobilov

3.12 AUTOŠKOLA

Zadanie: Databáza pre autoškolu slúži na uchovávanie dát o žiakoch, inštruktoroch a jednotlivých jazdách alebo teoretických vyučovaniach. Dôležitými dátami o žiakoch a inštruktoroch sú kontaktné údaje, ale aj dátum narodenia. Inštruktor navyše musí mať dátum nástupu do práce. Žiak si vyberá dátum vyučovania a v prípade jazdy aj vozidlo. Okrem toho musí žiak za vodičský kurz zaplatiť, pričom si môže zvoliť niektorý z bežných spôsobov platby. Evidujú sa tiež informácie o skúškach, ako jej dátum, žiaci ktorí ju absolvujú a ich výsledky testov a jazd.

Pomôcka: Medzi vyučovania nepatria len jazdy, ale aj teoretická výučba. Preto vyučovaniu nemusí byť vždy priradené vozidlo. Dátum a čas, kedy prebehla platba je možné evidovať prostredníctvom dátového typu timestamp, ktorý automaticky vygeneruje čas a dátum v momente vytvorenia prípadne upravenia záznamu.

Riešenie:



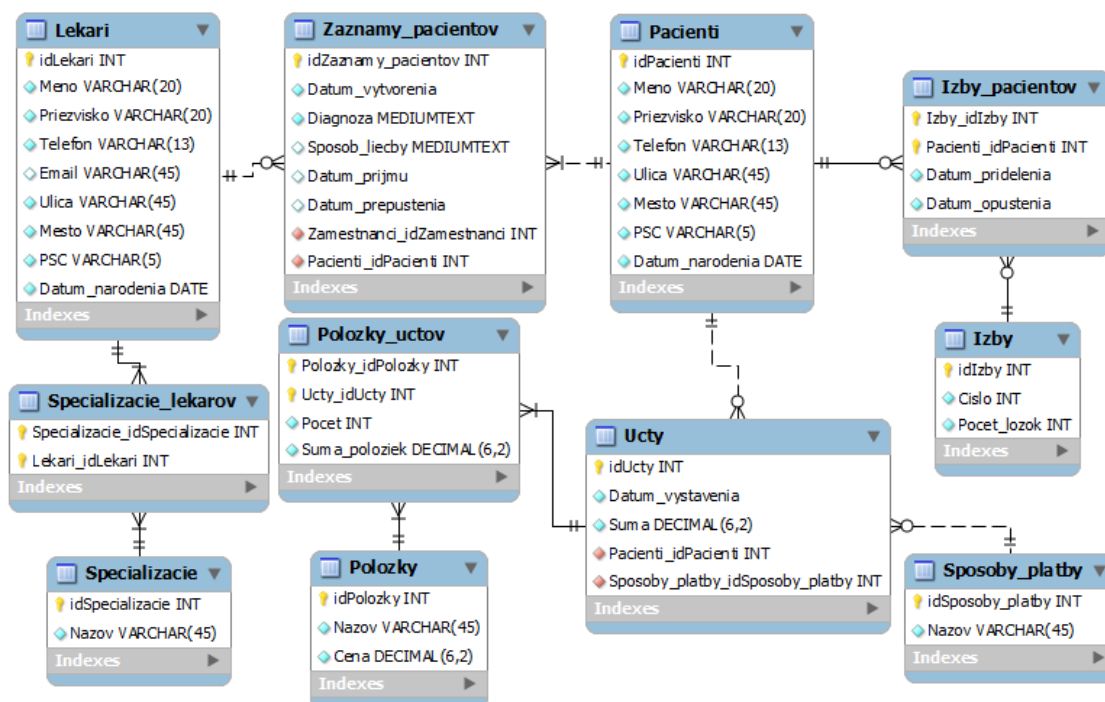
Obr. 20 Dátový návrh evidencie študentov autoškoly

3.13 NEMOCNICA

Zadanie: Cieľom je navrhnuť databázu, ktorá bude evidovať pacientov nemocnice a ich záznamy. Pacienti uvádzajú základné údaje vrátane dátumu narodenia. Každý pacient má vytvorený záznam o diagnóze a prípadnom spôsobe liečby. Ak bol hospitalizovaný, zaznamenáva sa aj dátum prijatia a prepustenia. Tieto údaje zapisuje lekár, o ktorom sa uchováva rovnaké údaje ako o pacientoch a navyše aj jeho špecializácie. Ak je pacient hospitalizovaný je nutné evidovať informácie o jemu pridelennej izbe, ako je číslo, počet lôžok, dátum pridelenia a opustenia danej izby. Pri opustení nemocnice môže byť pacientovi vystavený účet, ak mu boli vykonané spoplatnené úkony. Okrem spôsobu platby sa uvádza aj dátum vystavenia, celková suma a jednotlivé položky účtu. Položky a ich cena musia byť v databáze zaznamenané osobitne. Cena položiek sa môže časom meniť, tak je potrebné uchovávať cenu položiek aj pri jednotlivých účtoch.

Pomôcka: Môžu existovať lekári, ktorí ešte nemuseli vyplňať záznamy pacientov, a preto je účasť záznamov tabuľky Zaznamy_pacientov na vzťahu s tabuľkou Lekari voliteľná. Pacient nemusí byť hospitalizovaný a vtedy mu nie je pridelená žiadna izba. Podobne nemusí byť pacientovi vystavený blok v prípade, že úkony boli bezplatné.

Riešenie:



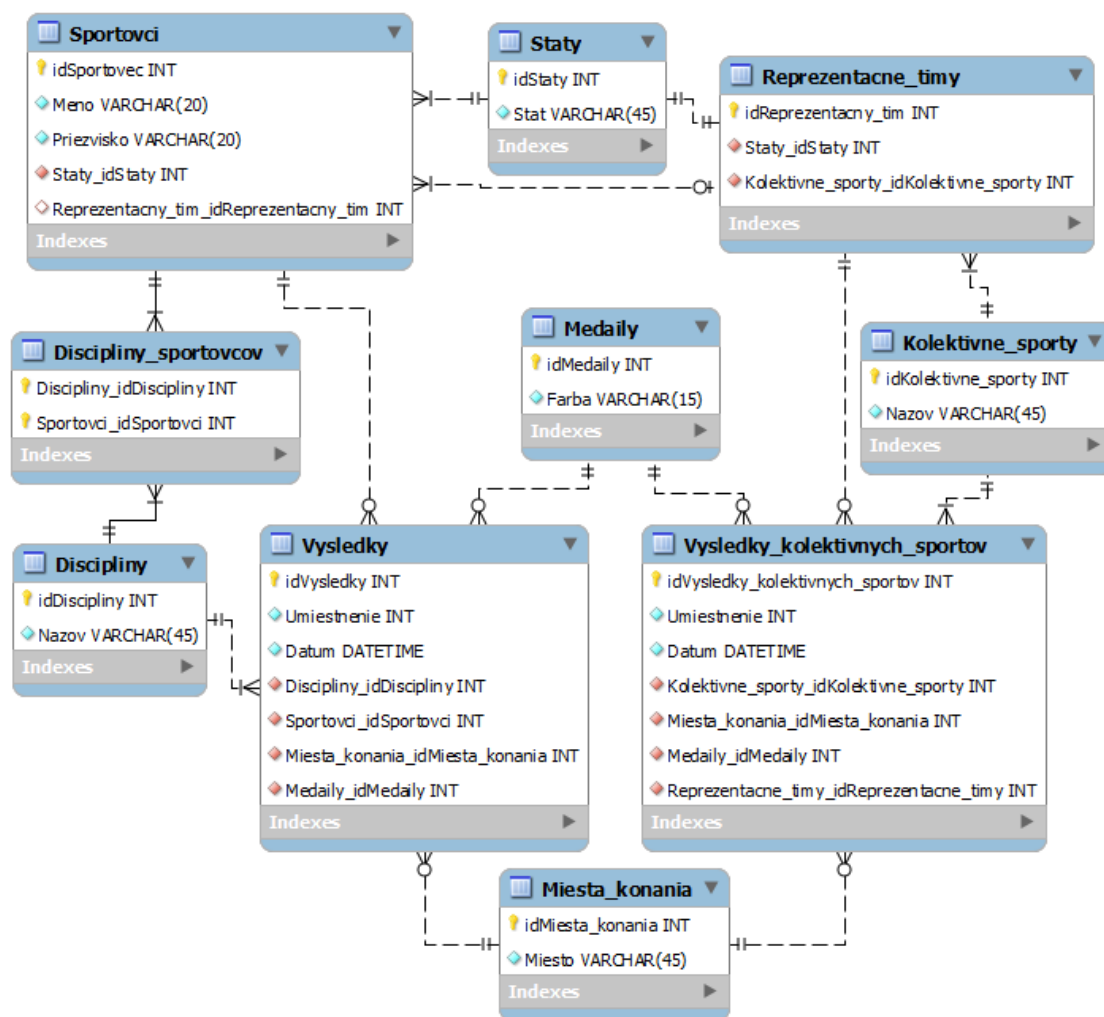
Obr. 21 Dátový návrh nemocnice

3.14 EVIDENCIA OLYMPIJSKÝCH HIER

Zadanie: Databáza eviduje výsledky jednotlivých športov na olympijských hrách. Športovec musí zastupovať niektorú krajinu a v prípade, že súťaží v kolektívnom športe, musí byť zaradený do reprezentačného tímu. Výsledky obsahujú umiestnenie športovca alebo tímu, ale aj dátum konania a disciplínu. Pri olympijských hrách je dôležité udeľovanie medaily. Keďže sa všetky disciplíny neodohrávajú na jednom mieste, je potrebné zaznamenať aj miesto konania.

Pomôcka: Športovec, ktorý sa zúčastňuje hier ako jednotlivec, nemusí byť zaradený do tímu súťažiaceho v kolektívnom športe.

Riešenie:



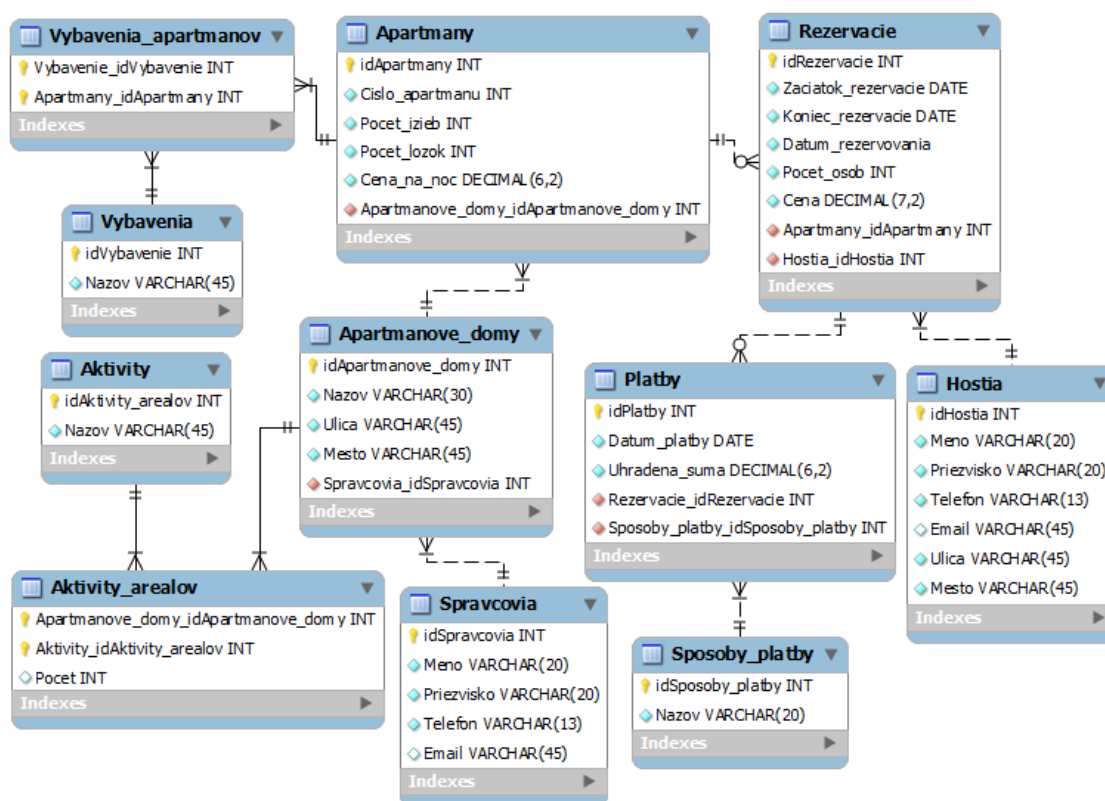
Obr. 22 Dátový návrh evidencie olympijských hier

3.15 UBYTOVANIE V APARTMÁNOCH

Zadanie: Cieľom tejto úlohy je vytvoriť databázu, ktorá umožňuje rezervovať ubytovanie v apartmánovom dome. Pre jednotlivé apartmány je potrebné uvádzať ich číslo, počet izieb, počet lôžok, cenu na noc a vybavenie. Vybavením je myslené pripojenie k wifi, satelitná televízia či kuchynské elektrospotrebiče. Vzhľadom na to, že apartmány sa nachádzajú v apartmánových domoch, je o týchto treba uchovávať názov, adresu, meno správcu a ostatné kontaktné údaje. Vhodné je uviesť aj aktivity, ktoré apartmánový dom poskytuje, ako napríklad minigolf, sauna, bowling a iné. Hostia pri rezervácii zadávajú svoje údaje a požadovaný začiatok a koniec ubytovania. Dôležitý je aj počet osôb a celková cena za rezerváciu. Eviduje sa i uhradená suma, spôsob a dátum platby. Platba sa uskutočňuje vo forme zálohy a zvyšok sa dopláca pri príchode.

Pomôcka: Medzi apartmánmi a rezerváciami vzniká vzťah s voliteľnou účasťou, lebo apartmán nemusí mať nevyhnutne rezerváciu.

Riešenie:



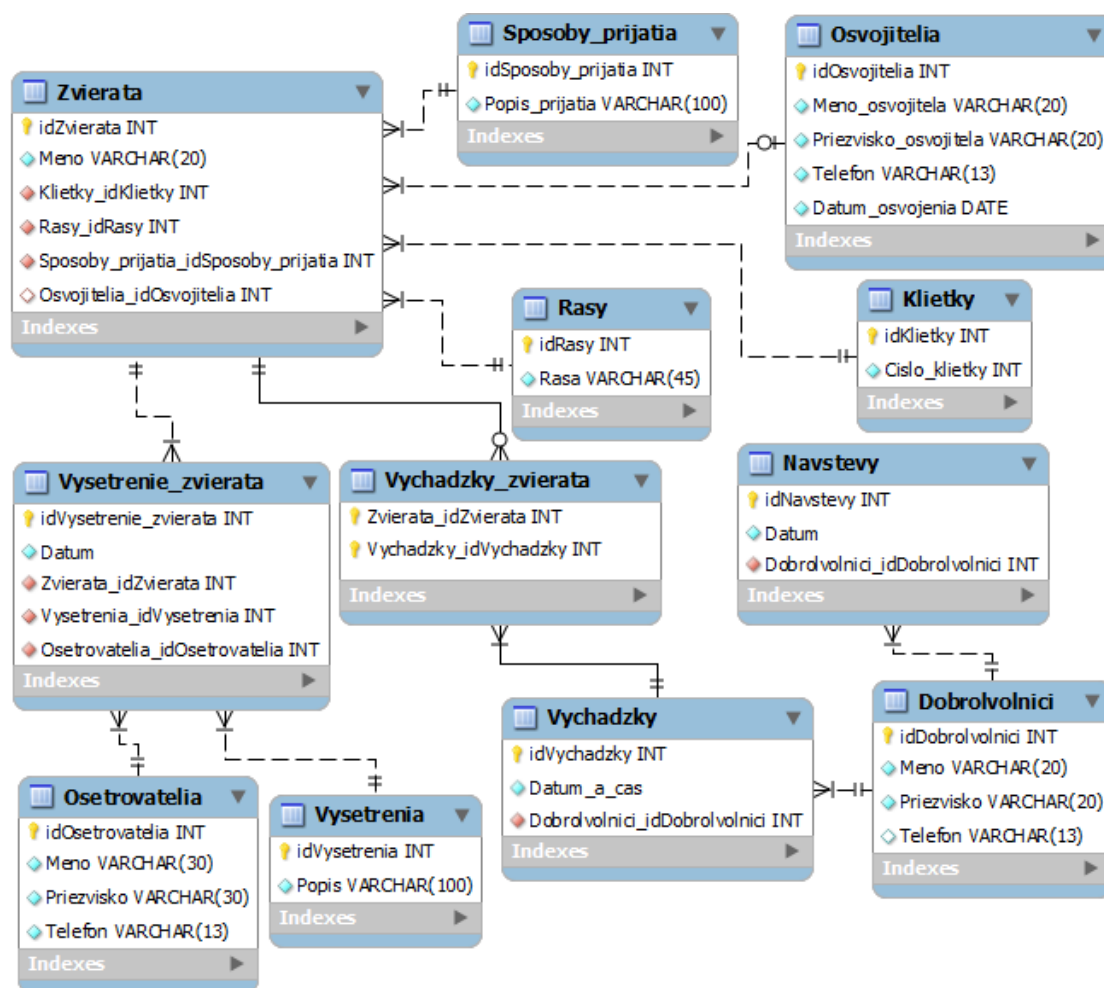
Obr. 23 Dátový návrh ubytovaní v apartmánoch

3.16 PSÍ ÚTULOK

Zadanie: Evidencia zvierat v psom útulku zahŕňa ich meno, rasu, číslo kliečky v ktorej sa nachádzajú a popis spôsobu ako sa do útulku dostali. Databáza obsahuje mená a príchody dobrovoľníkov, ktorí chodia so psami na vychádzky, ako aj čas týchto vychádzok. V každom útulku sa dbá o zdravie zvierat a preto sa pravidelne vyšetrujú. Pre každého psa sa vedie záznam o dátume a dôvode ošetrenia, a taktiež o ošetrovateľovi vykonávajúcom daný úkon. Cieľom útulku je nájsť pre psa nových majiteľov. Okrem mena osvojiteľa sa v databáze uchováva aj dátum osvojenia.

Pomôcka: Nie každý pes musí mať osvojiteľa a preto môže mať cudzí kľúč osvojiteľa v tabuľke Zvierata hodnotu null. Dôležitým údajom je, že jeden dobrovoľník môže na vychádzku vziať viacero psov naraz.

Riešenie:



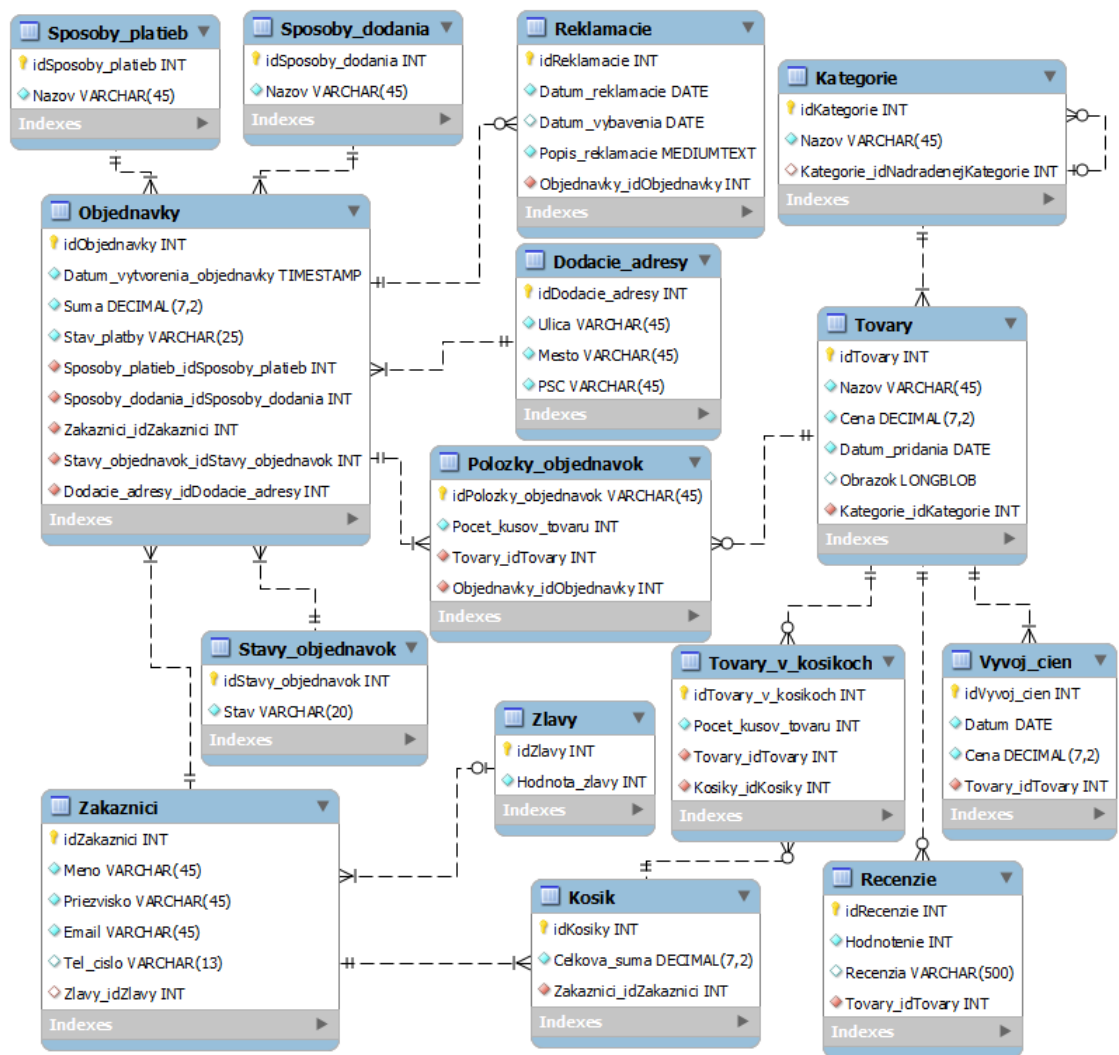
Obr. 24 Dátový návrh psieho útulku

3.17 ESHOP

Zadanie: Databáza umožňuje uchovávať dáta o zákazníkoch a ich objednávkach. Zákazníci pri registrácii zadávajú svoje kontaktné údaje a adresu, kde bude tovar dodaný. Verným zákazníkom sú poskytované zľavy. Zákazník si tovar pridáva do košíka, v ktorom sa uvádza celková cena a počet kusov daného tovaru. Tovar sa dá triediť podľa kategórií, pričom kategória môže byť rozdelená na ďalšie kategórie. K tovaru sa zobrazuje dátum pridania, cena a môže mať aj obrázok. Existuje možnosť pozrieť si recenzie a vývoj cien daného tovaru. Každá objednávka eviduje celkovú cenu, spôsob a stav platby, ako aj zvolený spôsob dodania. Zapisuje sa aj stav objednávky, ktorý uvádza či je tovar na ceste, alebo sa ešte len pripravuje. Súčasťou každého eshopu by mala byť možnosť reklamácie, ktorá obsahuje dátum a dôvod jej podania a tiež dátum vybavenia.

Pomôcka: Na rozdelenie kategórií do ďalších podkategórií je potrebné použiť rekurzívny vzťah, čiže tabuľku prepojiť samu so sebou. Vzniká tu niekoľko voliteľných vzťahov a preto si treba uvedomiť, že každá objednávka nemusí mať reklamáciu, kategória nemusí mať podkategóriu, tovar nemusí byť v košíku, tovar nemusí mať recenziu, košík nemusí obsahovať tovar, tovar nemusí byť položkou objednávky a zákazník nemusí mať zľavu.

Riešenie:



Obr. 25 Dátový návrh eshopu

ZÁVER

Cieľom bakalárskej práce bolo vytvoriť materiál, ktorý by poskytol čitateľom návod ako vytvárať návrhy relačných databáz. Takéto databázy sa využívajú v obchodoch, bankách, letiskách ale aj v mnohých ďalších odvetviach. Vďaka tomu firmy zaoberajúce sa tvorbou databáz ponúkajú množstvo pracovných pozícií ľuďom so znalosťami v tomto obore.

Bakalárska práca sa skladá z teoretickej a praktickej časti, ktoré majú čitateľa motivovať a oboznámiť ho s touto problematikou. Teoretická časť zahŕňa vysvetlenie základných pojmov a prvkov relačných databáz. Zaoberá sa rozdelením dátových modelov a predovšetkým charakterizuje entitno-relačný model. Venuje sa tiež popisu anomálií, ktoré môžu vzniknúť pri jeho návrhu a ich eliminovaniu pomocou normalizácie, ktorá je zhrnutá v metodike. Pri tvorbe tejto časti sme prekladali najmä zdroje z anglickej a českej literatúry.

Praktická časť pozostáva zo zbierky príkladov. Prvým príkladom je filmová databáza, na ktorej je rozpracovaný postup jej návrhu. V podrobne popísaných krokoch je uvedená tvorba tabuliek na základe identifikovaných entít, definovanie atribútov a primárnych kľúčov a nakoniec vytváranie vzťahov medzi tabuľkami. Samotný návrh je vytvorený v CASE nástroji MySQL Workbench, ktorý je určený na tvorbu takýchto návrhov.

Zvyšné príklady tiež obsahujú riešenie vo forme výsledného návrhu databázy, ale neobsahujú už podrobný postup jeho tvorby. Každý príklad však obsahuje pomôcku, ktorá má čitateľovi pomôcť pochopiť niektoré časti návrhu. Zadania v zbierke sú zoradené tak, aby so stúpajúcou náročnosťou kládli väčšie požiadavky na čitateľa a tým zdokonalili jeho zručnosti v tejto oblasti. Inšpiráciou pri výbere príkladov bola stránka www.databaseanswers.org.

Verím, že táto bakalárska práca bude prínosom pre všetkých, ktorí sa chcú venovať tvorbe návrhov relačných databáz.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

- CONNOLLY, T. M. - BEGG, C. E. 2005. Database Systems: A Practical Approach to Design, Implementation, and Management. Havlíčkův Brod: Addison-Wesley, 2005. 1424 s. ISBN 978-0-32121-025-8.
- CORONEL, C. – MORRIS, S. 2014. Database Systems: Design, Implementation, & Management. Stamford, Connecticut: Cengage Learning, 2014. 1425 s. ISBN 9781305323230.
- DRLÍK, M. - SKALKA, J. 2007. Databázové systémy - prednášky. [Online]. 2014. [cit. 2014-04-05]. Dostupné na internete: <<https://edu.ukf.sk/mod/resource/view.php?id=64832>>.
- FORTA, B. 2011. MariaDB Crash Course. Crawfordsville, Indiana: Addison-Wesley Professional, 2011. 304 s. ISBN 978-0-13-284235-8.
- GILMORE, W. J. 2010. Beginning PHP and MySQL: From Novice to Professional. New York: Apress, 2010. 824 s. ISBN 978-1-4302-3114-1.
- GRAVELLE, R. 2010. Cardinality in MySQL Data Modeling. [Online]. 2014. [cit. 2014-04-05]. Dostupné na internete: <<http://www.databasejournal.com/features/mysql/article.php/3906761/Cardinality-in-MySQL-Data-Modeling.htm>>.
- HERNANDEZ, M. J. 2006. Návrh databází. Havlíčkův Brod: Grada Publishing, 2006. 408 s. ISBN 80-247-0900-7.
- HOPKINS, D. C. 2010. Elements of Relational Database Design. Victoria, British Columbia, Kanada: Trafford Publishing, 2010. 128 s. ISBN 978-1-4251-8280-9.
- MEMBREY, P. - PLUGGE, E. - HAWKINS, T. 2010. The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing. New York: Apress, 2010. 328 s. ISBN 978-1-4302-3052-6.
- MERSON, P. 2009. Data Model as an Architectural View. [Online]. 2014. [cit. 2014-02-05]. Dostupné na internete: <<http://www.sei.cmu.edu/reports/09tn024.pdf>>.
- OPPEL, A. 2006. Databáze bez předchozích znalostí. Brno: Computer Press, 2006. 319 s. ISBN 80-251-1199-7.

- OPPEL, A. 2012. SQL bez předchozích znalostí. Brno: Computer Press, 2012. 240 s. ISBN 978-80-251-1707-1.
- RICARDO, C. 2011. Databases Illuminated. Sudbury, Massachusetts: Jones & Bartlett Publishers, 2011. 661 s. ISBN 978-1-4496-0600-8.
- RIGGS, S. - KROSING, H. 2010. PostgreSQL 9 Administration Cookbook: Solve Real-world PostgreSQL Problems with Over 100 Simple, Yet Incredibly Effective Recipes. Birmingham: Packt Publishing Ltd, 2010. 345 s. ISBN 978-1-84951-029-5.
- SINGH, S. K. 2011. Database Systems: Concepts, Design and Applications. Noida, India: Pearson Education India, 2011. 944 s. ISBN 978-81-317-6092-5.
- SPOIALA, C. 2013. CouchBase vs CouchDB vs MongoDB. [Online]. 2014. [cit. 2014-04-05]. Dostupné na internete: <<http://assist-software.net/blog/couchbase-vs-couchdb-vs-mongodb>>.
- SUMATHI, S. - ESAKKIRAJAN, S. 2007. Fundamentals of Relational Database Management Systems. Berlin, Heidelberg: Springer Science & Business Media, 2007. 776 s. ISBN 978-3-540-48397-7.
- WILLIAMS, B. 2014. Data Models. [Online]. 2015. [cit. 2015-02-17]. Dostupné na internete: <http://www.databaseanswers.org/data_models/>.

ZOZNAM PRÍLOH

Príloha A – CD nosič obsahuje:

- Bakalárska práca vo formáte pdf,
- návrhy relačných databáz vo formáte mwb.