INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

DUM 10 téma: Relační databázový systém

ze sady: 3 **tematický okruh sady:** III. Databáze
ze šablony: 7 – Kancelářský software **určeno pro:** 4. ročník
vzdělávací obor: 18-20-M/01 Informační technologie
vzdělávací oblast: odborné vzdělávání
metodický list/anotace: viz VY_32_INOVACE_07310ml.pdf

Databáze

Hlavním úkolem oboru informačních technologií je shromažďovat, uchovávat a následně prezentovat informace. To činí nejčastěji prostřednictvím zpracování obrovských množství dat – tedy kvantifikovatelných popisů skutečnosti. Teprve data vhodně uložená, dohledatelná a především pro uživatele informačního systému užitečná nazýváme „informace“.

Jakoukoliv aplikace informačních systémů, která disponuje uvedenými funkcemi pak označujeme jako databáze, nebo databázový systém.

Typy databázových systémů

Historickým vývojem prošla řada technik a metod přehledného ukládání dat, mnohé převzaté z doby před-počítačové, kdy ovšem informatická věda již dávno existovala. Pro čtenáře bude možná tento fakt překvapující, ale i v letech před sestrojením prvních elektronických počítačů lidstvo data shromažďovalo, třídilo a vyhledávalo. Činnost tato byla jistě pracnější, časově a lidsky náročnější, na druhou stranu však tato pracnost nutila vybírat opravdu pouze užitečné informace a omezovat jejich množství na nezbytné minimum. V dnešní době tato bariéra mnohdy padá, a skladujeme tak často pouze data – bez informační hodnoty. Svůj pohled na tento vývoj, nechť si laskavý čtenář zaujme individuálně.

Starší databázové systémy stručně představíme, nebudeme se jimi však blíže zabývat [1]:

Základním databázovým modelem byla prostá „halda“ dat v jednom či několika souborech. Hovoříme pak o souborové databázi. Informační technologie sloužily pouze jako datový nosič, zbylá inteligence – k udržování konzistence obsahu, vzájemných vazeb, bezpečnosti a především vyhledávání zůstávala v rukou uživatele. Takový model nazýváme souborovým modelem FMS.

Následují hierarchické a sekvenční databázové modely, které již abstrahují od závislosti na konkrétním způsobu uložení dat na datovém médiu a zavádí pojem struktury, kterou již zná přímo databázový systém – a jako takový ji dokáže kontrolovat a vynucovat. Jednotlivé prvky databáze jsou spojeny na různých úrovních a rozeznáváme i vztahy rodičovské, či nadřazenosti. Výhodou (oproti souborovým databázím) je možnost rychlého nalezení požadované informace, pouze při znalosti počátečního bodu hledání (úvodního prvku, kořene atd.) a směru či omezujících podmínek. Zde se již projevily výhody zpracování pomocí prostředků informačních technologií, protože ruční realizace takového systému již prakticky není možná.

Zásadní nevýhodou obou modelů je de-facto nemožnost jakkoliv měnit (tedy i rozšiřovat) strukturu již provozovaného systému.

Relační systém

Zásadním stupněm vývoje, kterému se budeme věnovat nejvíce, je relační databázový systém. Jeho autorem je především doktor E.F.Codd a jeho první charakteristiku představil již v roce 1969 [1]. Ten v sobě spojuje výhody předešlých způsobů ukládání dat, dovedený do vysokého stupně formality prostřednictvím využití matematické teorie množin.

Na data v něm uložené nahlížíme jako na prvky různých množin a vztahy mezi nimi (odtud „relace“) popisujeme jako kartézské součiny těchto množin. Můžeme tak velmi intuitivně použít i pojmy „zobrazení“ z matematiky.

V relačním systému vytváříme model databáze nejčastěji pomocí formálně grafických metod s využitím řady diagramů. Pro oblast středoškolské praxe vystačíme pouze se statickým modelem struktury, obsahujícím Entity a Relace .. odtud E-R diagram (ERD). Dalšími částmi modelu jsou DFD, či STD a podobně.

Entity a relace

Entita

Je modelem prvku reálného světa, který se navenek vyznačuje určitými vlastnostmi a vyskytuje se ve více, vzájemně odlišitelných, exemplářích. Velmi zjednodušeně lze říci, že entitou je vše – co můžeme označit podstatným jménem.

Entitou je pak student, člověk, automobil, škola a podobně. Jednotlivé vlastnosti entity označujeme jako atributy. Přitom reálné entity mají vždy prakticky nekonečné množství atributů a je úkolem autora databázového modelu vybrat pouze ty, které jsou pro model podstatné.

Například při tvorbě modelu knihovny, kde se vyskytují i čtenáři, nás u nich bude zajímat bydliště či telefonní číslo. Ovšem velikost bot nebo barva očí bude pro takový model nepodstatná.

Zvláštní vlastností entit je i možnost jejich vzájemné hierarchie, kdy třeba u modelu školy mají entity student a učitel řadu vlastností společných – spojitelných do entity člověk (jméno, bydliště…) a teprve specifické vlastnosti pak budeme sledovat zvlášť u studenta

a učitele (například studovaný ročník a naopak číslo kabinetu). Modely s vnitřní hierarchií však přenecháme až vyšším stupňům studia.

Podle charakteru atributů entit pak můžeme rámcově rozlišit jejich datové typy – číselné, textové, datumové… Přesné rozlišení je však závislé na konkrétním databázovém systému, pro který model vytváříme.

Atributy, které lze použít pro jednoznačné rozlišení jednotlivých výskytů téže entity (hovoříme o instancích) nazýváme kandidátskými atributy. U takových atributů především požadujeme, aby byl přítomen u každého výskytu entity bez výjimek. Dále aby u všech entit byla jeho hodnota unikátní a neměnná. Příklad – u entity automobil by jako kandidátský atribut přicházelo do úvahy VIN číslo nebo výrobní číslo karoserie (u starších vozů), naopak naprosto nevhodným by byla barva nebo počet dveří. U osob je problematika jejich jednoznačné identifikace složitější, otázka možného využití rodného čísla a naopak snahy omezit jeho využívání je mimo rozsah tohoto materiálu.

Jeden z vybraných kandidátských atributů pak nazýváme primárním klíčem a pro každou entitu musíme nejméně jeden takový v modelu určit.

Relace

Popisuje vztahy mezi entitami vzájemně a to jak jejich prostou existenci, tak i jejich kvalitu – míru. Základním předpokladem relačního modelu je existence vzájemných vztahů mezi entitami, a naopak přítomnost entity bez jakéhokoliv vztahu k ostatním by měla být pro autora modelu signálem, že taková entita je v modelu vlastně zbytečná. Výjimkou jsou samozřejmě jednoentitní „databáze“.

Z hlediska kvality vztahů pak hovoříme o kvantifikovanosti relací – zjednodušeně o počtu instancí jednotlivých entit na obou stranách vztahu. Rozlišujeme pak:

- relaci 1:1 – tzv. rozšiřující, kdy vlastně jednu entitu přímo propojujeme s konkrétní další
- relaci 1:N – tzv. identifikační, kde jeden výskyt entity je ve vztahu k několika výskytům entity jiné
- relaci M:N – tzv. volná, kde na obou stranách najdeme několik výskytů obou entit

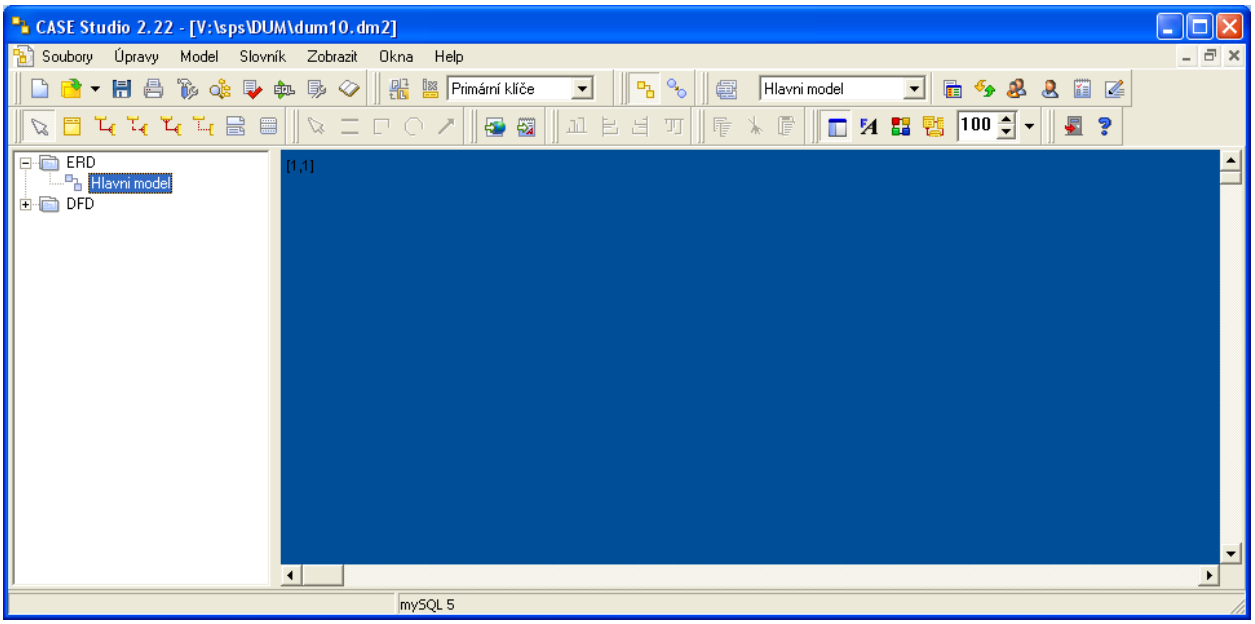
Příklady:

- Student:škola – 1:N – student studuje jen na jedné škole, škola ovšem vzdělává několik studentů
- Předmět:student – M:N – předmět má zapsáno několik studentů, každý student ale zároveň studuje více různých předmětů
- Automobil:motor – 1:1 – v automobilu je právě jeden motor a motor je právě v jednom automobilu

V databázové praxi jsou naprosto nejčastější relace typu 1:N, přičemž relace M:N většina databázových systémů neumí přímo modelovat (viz dále) a relace 1:1 většinou vedou ke spojení dvou entit do jedné.

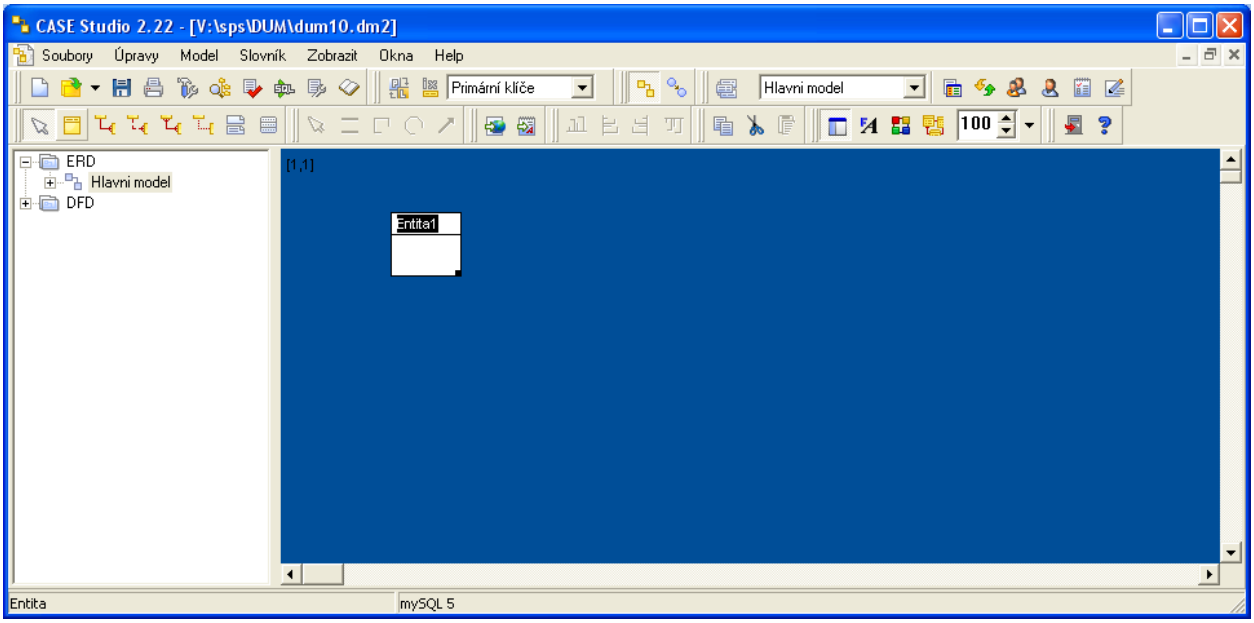
Databázový model

Pro ukázku tvorby databázového modelu použijeme modelovací nástroj (CASE) typu CaseStudio:

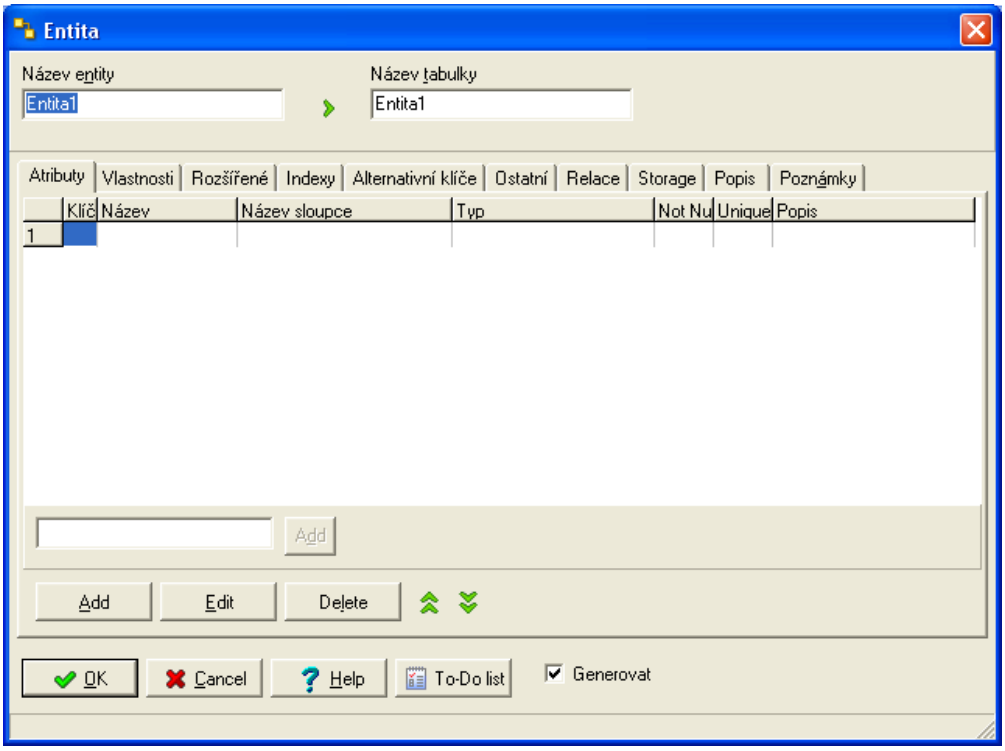


Vidíme pracovní prostor modelu, vlevo strom diagramů a otevřený diagram ERD „Hlavní model“. U rozsáhlých modelů můžeme jednotlivé části rozdělit do konceptuálních úrovní, což je ale mimo rozsah látky střední školy.

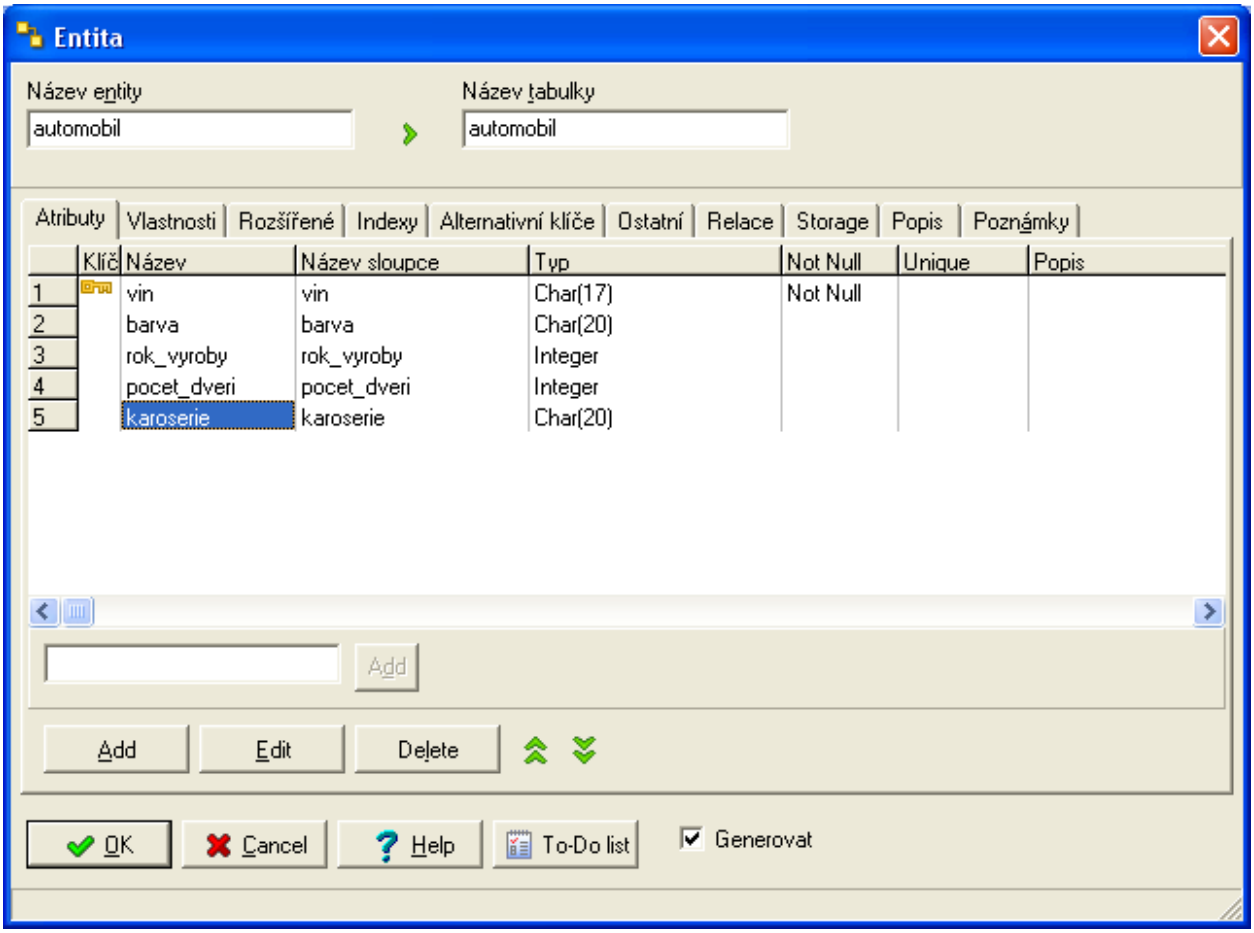
Vytvoříme první entitu:



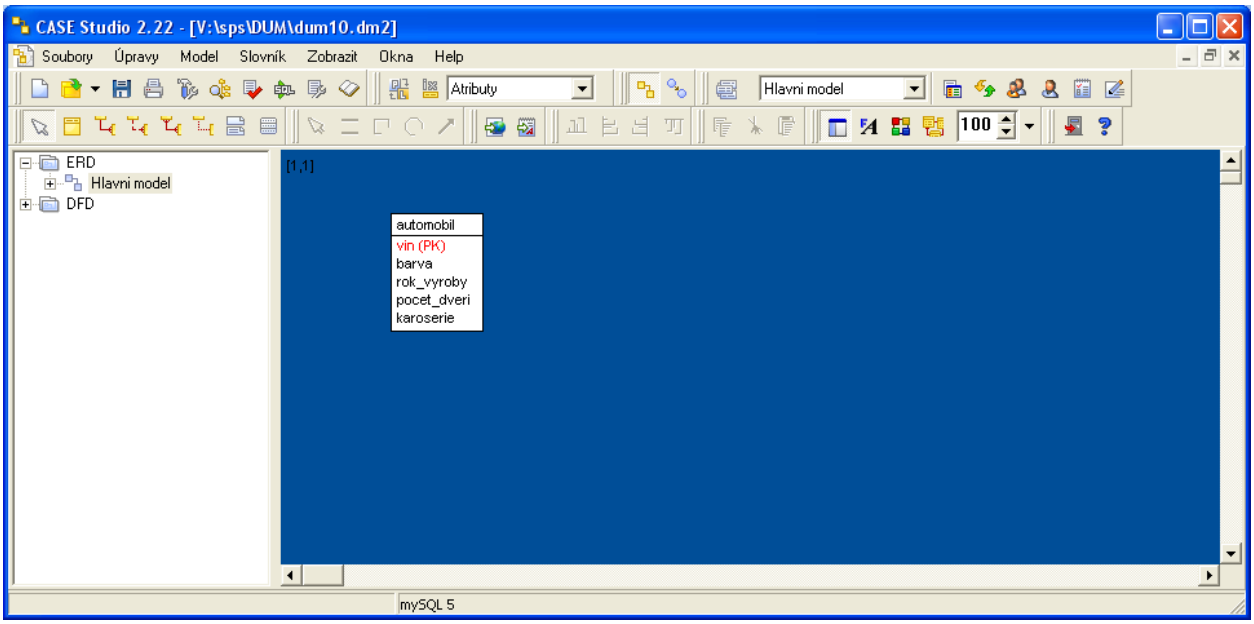
a nastavíme její vlastnosti:



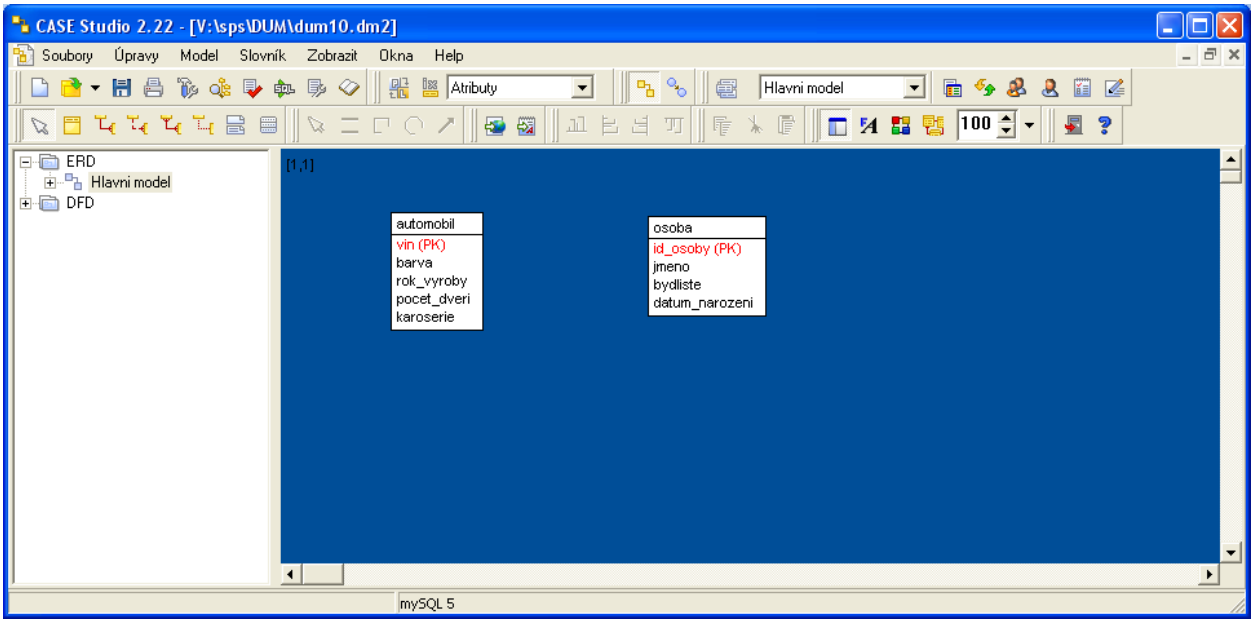
zde je vhodné zvolit stručný ale výstižný název entity – bez diakritiky, mezer a nejlépe velkých znaků. Dále entitě doplníme i atributy:



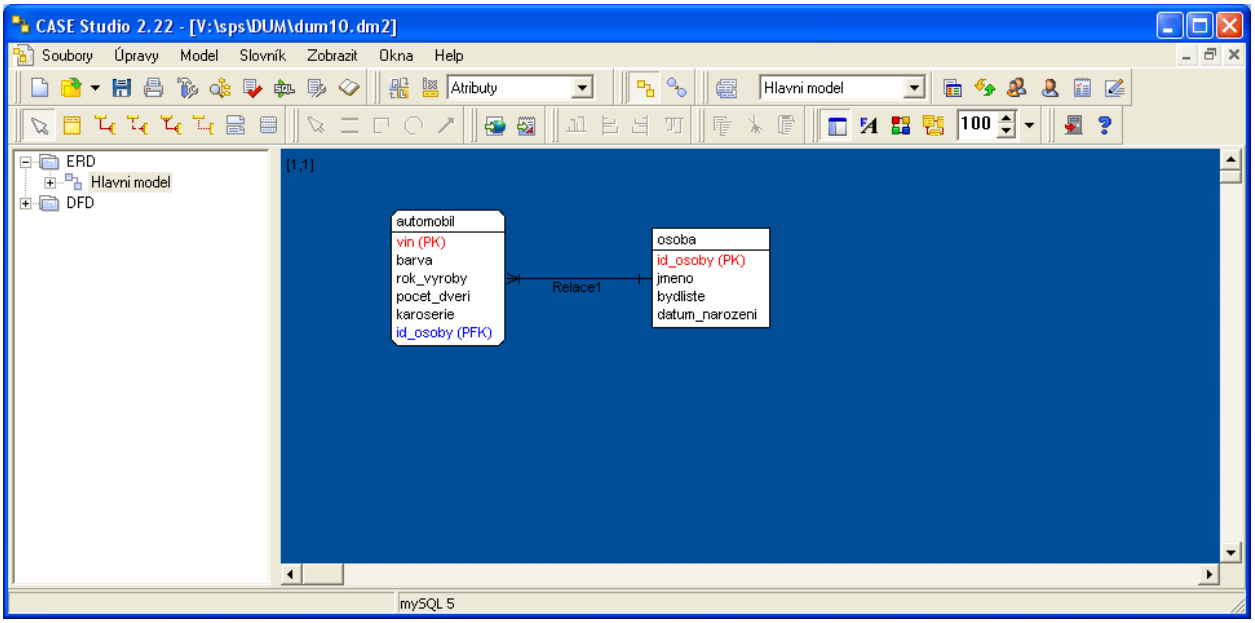
U zvoleného atributu nastavíme i parametr primárního klíče. Entita se pak zobrazí v prostoru diagramu:



Doplníme stejným způsobem i druhou entitu:



A namodelujeme relaci. Předpokládáme, že automobil má vždy jediného vlastníka a naopak osoba vlastní libovolné množství automobilů. Jedná se tedy o relaci 1:N. Pro modelování využijeme primární klíč (PK) u entity osoba, který propojíme s novým cizím primárním klíčem (FPK) u entity automobil. Entita automobil tak získá nový atribut, ve kterém bude odkaz na osobu která ho vlastní. Hierarchicky je tak entita automobil podřízena entitě osoba:



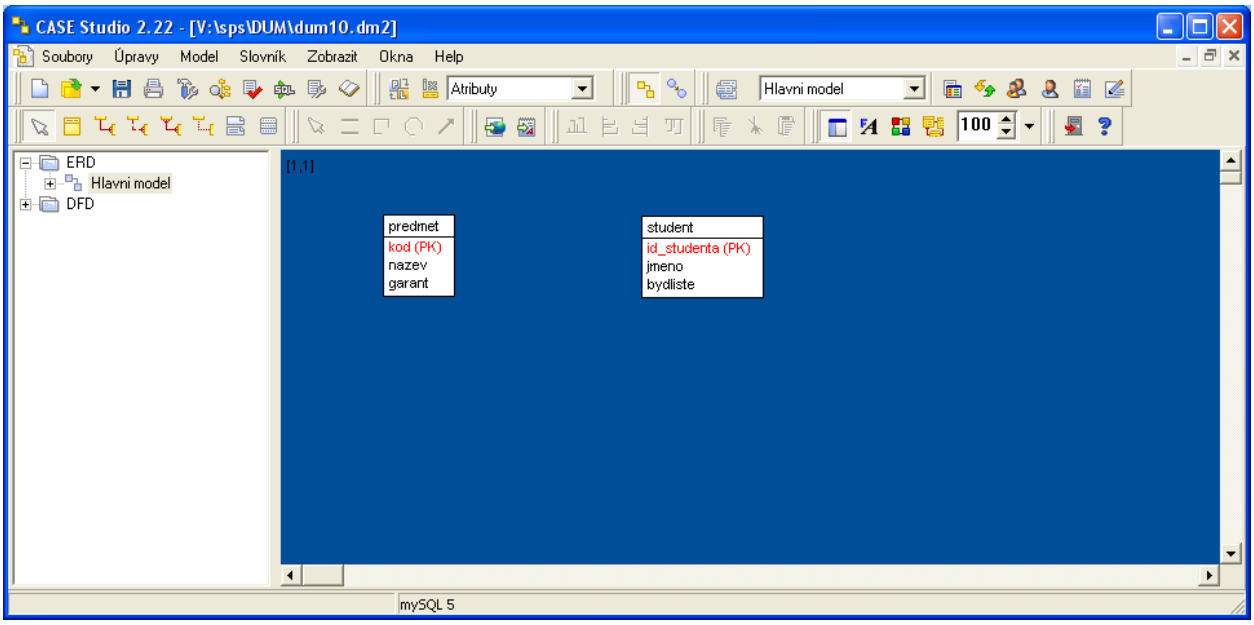
Grafický element entity automobil na to zareagoval i změnou svého vzhledu. FPK u entity automobil můžeme i přejmenovat, v našem případě to však není nutné. Současný název je logický. Zde se i ukazuje vhodnost používání unikátních názvů atributů pro primární klíče

v celém modelu – odpadnou nejasnosti v situacích, kdy by se ve všech entitách jmenovaly „id“ a podobně.

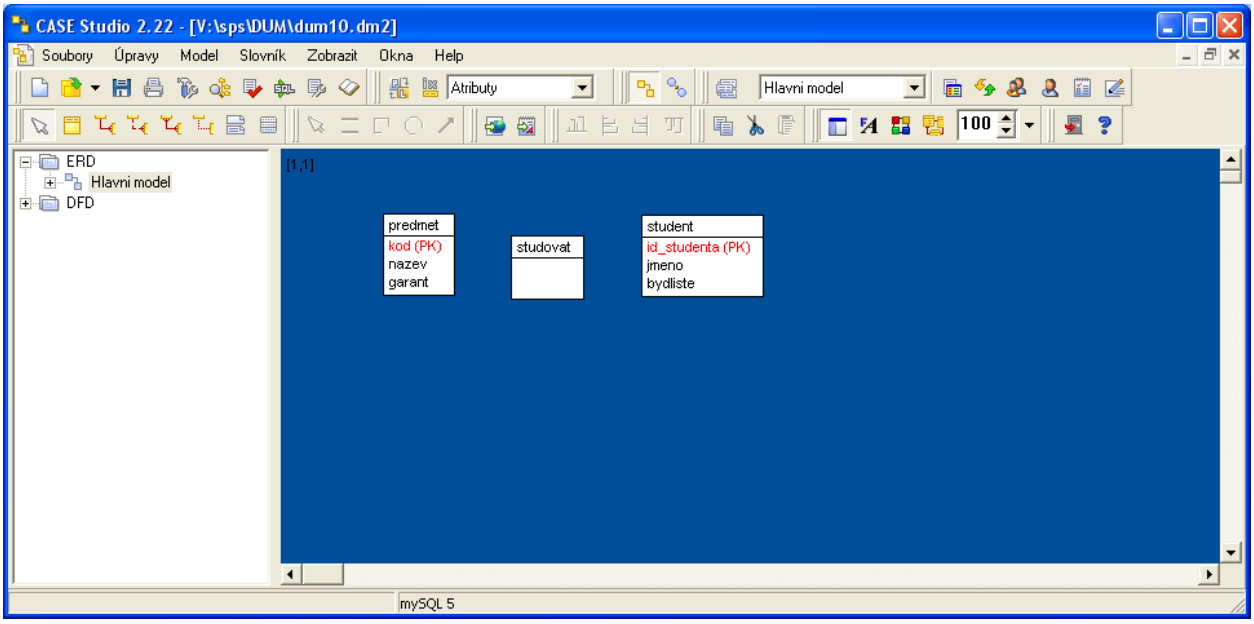
Modelování relace M:N

Tento typ relace nelze přímo modelovat a model musíme rozšířit o pomocnou virtuální („vazební“) entitu. V další části ji pak budeme nazývat vazební (relační/spojovací) tabulka.

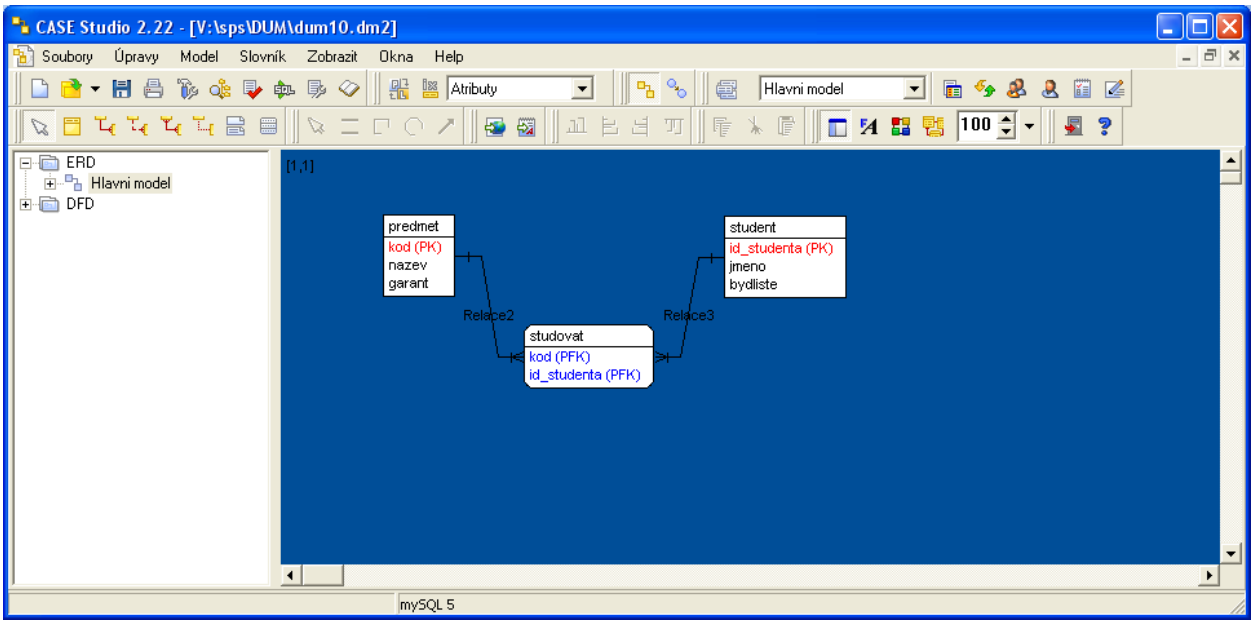
Rozdělíme tak relaci M:N na dvě dílčí 1:N. Příklad:



Mezi entitami student a predmet chceme realizovat relaci M:N. Vložíme tedy pomocnou entitu „studovat“, která nebude obsahovat žádné atributy:



Nyní pomocí PK kod a PK id_studenta obě entity propojíme v pomocné entitě:



Entita studovat teď obsahuje pouze FPK. Každá její instance bude popisovat vztah konkrétního studenta a konkrétního předmětu. Přitom každý student může být účastníkem libovolného počtu těchto instancí (studovat libovolný počet předmětů) a naopak každý předmět může vystupovat ve více těchto vazbách – být studován více studenty.

Úkoly pro samostatnou práci

Sestavte ERD pro modely:

- knihovna
- autobazar
- zdravotnické zařízení
- střední škola

Zdroje:

- Archiv autora
- [1] KOCAN, Marek. Databáze nejsou jen relační: díl nultý. *Databázový svět* [online]. 2001 [cit. 15.1.2013]. Dostupné z: http://www.dbsvet.cz/view.php?cisloclanku=2001111201