

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Databázově koncipované informační zabezpečení zvyšování
kvalifikace zaměstnanců

Tomáš Trnka

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Trnka

Informatika

Název práce

Databázově koncipované informační zabezpečení zvyšování kvalifikace zaměstnanců

Název anglicky

Database concept securing the qualification improvement of employees

Cíle práce

Bakalářská práce je zaměřena na problematiku databázově koncipované informační podpory procesu zvyšování kvalifikace zaměstnanců v podnikové sféře. Hlavním cílem této práce je:

- objasnit teoretické principy relačně databázové technologie, datového modelování, datové normalizace a integrity v kontextu s problematikou informačního zabezpečení zvyšování kvalifikace zaměstnanců,
- zmapovat momentální stav této problematiky a vymezit její relevantnost včetně požadavků na ní kladených,
- navrhnout přijatelné řešení této problematiky v souladu s identifikovanými požadavky,
- ověřit funkčnost navržených záležitostí,
- ověřené záležitosti zobecnit pro další možná uplatnění.

Metodika

Použitá metodika této bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Navrhované řešení bude zohledňovat identifikované požadavky a očekávání spojená s řešenou záležitostí. Na podkladě syntézy teoretických poznatků a dosažených výsledků budou formulovány závěry této bakalářské práce a následně zobecněny pro další možná použití.

Doporučený rozsah práce

45-55 stran

Klíčová slova

relačně databázová technologie, datové modelování, datová normalizace a integrita, informační potřeba, zvyšování kvalifikace zaměstnanců

Doporučené zdroje informací

BEGG, C., CONOLLY, T., HOLOWCZAK, R.: Mistrovství databáze, profesionální průvodce tvorbou efektivních databází. Computer Press. Brno 2009. ISSN 978-80-251-2328-7

BRYLA, B., LONEY, K.: Mistrovství v Oracle Database 10g. Computer Press Brno 2006. EAN 978802512779

HERMANDEZ, M.: Návrh databází, GRADA 2005. ISBN 80-247-0900-7.

LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2007. EAN 97880251149002.

LONEY, K.: Oracle Database, kompletní průvodce. Computer press Brno 2010. ISBN 978-80-251-2489-5

MOLINARO, A.: SQL Kuchařka programátora. Computer Press. Brno 2009. ISBN 978-80-251-2617-2

POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. ISBN 978-80-01-05212-9

Předběžný termín obhajoby

2015/06 (červen)

Vedoucí práce

doc. Dr. Ing. Václav Vostrovský, Ph.D.

Elektronicky schváleno dne 12. 3. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 16. 03. 2015

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Databázově koncipované informační zabezpečení zvyšování kvalifikace zaměstnanců" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 16.3.2015

Poděkování

Rád bych touto cestou poděkoval doc. Dr. Ing. Václavu Vostrovskému, Ph.D., za odborné vedení bakalářské práce a za cenné rady týkající se jejího zpracování.

Databázově koncipované informační zabezpečení zvyšování kvalifikace zaměstnanců

Database concept securing the qualification improvement of employees

Souhrn

Hlavním cílem této bakalářské práce je vytvořit databázový návrh pro aplikaci, která se zabývá procesem zvyšování kvalifikace zaměstnanců v podnikové sféře.

Kromě objasnění teoretických principů relačně databázové technologie je významnou částí definice požadavků na datový model, ovlivněna především současnými trendy v oblasti firemního vzdělávání.

Stěžejní částí je ovšem samotný návrh konceptuálního datového modelu odpovídající požadavkům stanovených v předchozí části. Přínosem je aplikování evidence zaměstnanců a exekuce online testů, resp. možnosti přihlášení ke kurzům v jednotném datovém modelu.

Summary

The bachelor work's main objective is to create a database concept concerning the process of improvement of employees' qualification in business sector.

Besides explaining the theoretical principles of relational database technology, there is an important part of defining the requirements for a data model, mainly influenced by current trends in corporate education.

The main part of the concept itself, is conceptual data model corresponding to the requirements specified in the previous section. The contribution of this work is the application of employee records and the execution of tests in a single data model.

Klíčová slova: relačně databázová technologie, datové modelování, datová normalizace a integrita, informační potřeba, zvyšování kvalifikace zaměstnanců

Keywords: relational database technology, data modeling, data normalization and integrity, information need, improvement of employees' qualification

Obsah

1 Úvod	4
1.1 Vymezení tématu práce a důvod jeho výběru.....	4
2 Cíl práce a metodika.....	5
2.1 Cíl práce.....	5
2.2 Metodika	5
3 Teoretická východiska	6
3.1 Databáze a její stručná historie	6
3.2 Databázový systém	6
3.3 Základní pojmy	8
3.4 Datová normalizace	13
3.4.1 První normální forma (1.NF)	14
3.4.2 Druhá normální forma (2.NF).....	15
3.4.3 Třetí normální forma (3.NF).....	15
3.4.4 Boyce-Coddova normální forma (BCNF)	16
3.5 Metodologie návrhu databáze.....	16
3.5.1 Konceptuální návrh databáze.....	16
3.5.2 Logický návrh databáze	20
3.5.3 Fyzický návrh databáze	20
4 Zmapování současného stavu a definice požadavků	21
4.1 Současný stav.....	21
4.2 Přínosy informačního systému pro vzdělávání zaměstnanců	22
4.3 Přístup firem ke zvyšování kvalifikace.....	22
4.4 Způsob definování požadavků	23
4.5 Typy a charakteristiky uživatelů.....	23
4.6 Základní požadavky na systém	25
4.7 Dílčí požadavky na systém	26
4.7.1 Formy školení	26
4.7.2 Struktura online testů	26
4.7.3 Struktura kurzů	26
4.7.4 Práva uživatelů.....	27
4.7.5 Práce s dokumenty	28
4.8 Pohledy uživatelů.....	28

5 Řešení konceptuálního návrhu	32
5.1 Správa osob.....	32
5.1.1 Entity a jejich atributy.....	33
5.1.2 Vztahy mezi entitami	35
5.2 Správa kurzů	36
5.2.1 Entity a jejich atributy.....	37
5.2.2 Vztahy mezi entitami	39
5.3 Správa testů.....	41
5.3.1 Entity a jejich atributy.....	41
5.3.2 Vztahy mezi entitami	43
5.4 Ověření a zobecnění funkčnosti pro další možná uplatnění	44
6 Závěr	45
7 Seznam použitých zdrojů	46
8 Seznam obrázků.....	47
9 Seznam tabulek	48
10 Příloha A – Konceptuální datový model.....	Chyba! Záložka není definována.

1 Úvod

Klíčovými zdroji každé firmy jsou její zaměstnanci, kteří musí splňovat určitou kvalifikaci. V současné situaci na trhu se vyskytuje silná konkurence, a právě dostatečná kvalifikace zaměstnanců pomáhá udržet krok organizace se stále častějšími změnami technologií. Kvalifikovanější pracovní síla dokáže přebírat složitější úkoly, což vede ke zvýšení efektivity a budoucímu růstu firmy^[6].

V dnešní době využívají větší organizace pro školení svých zaměstnanců specifické platformy, které umožňují přihlášení na prezenční kurzy či přímo exekuci e-learningových kurzů. Systém vzdělávání z dlouhodobého hlediska představuje nižší náklady na vzdělávání svých zaměstnanců a zároveň naplňuje jejich informační potřebu.

1.1 Vymezení tématu práce a důvod jeho výběru

Bakalářská práce je zaměřena na analýzu a samotnou tvorbu databáze, která bude splňovat mnou předem stanovené požadavky. Ve výsledku se jedná o vytvoření datových struktur a zřízení vztahů mezi tabulkami. Nejedná se o implementaci uživatelské aplikace.

Téma jsem si zvolil z důvodu osobního zájmu v oblasti osobního rozvoje, ale také kvůli možnosti využití teoretických poznatků z České zemědělské univerzity v rámci předmětu Databáze. Spojení návrhu databází a osobního rozvoje vyústilo ve snahu navrhnout konceptuální datový model pro aplikaci zabývající se evidencí a zvýšením kvalifikace zaměstnanců.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem je zmapovat aktuální potřebu řešené problematiky, navrhnout přijatelné databázové řešení v souladu s identifikovanými požadavky na model a v neposlední řadě ověřit funkčnost za pomoci zvoleného softwaru PowerDesigner.

Konečný návrh bakalářské práce bude uveřejněný na internetu a bude dostupný jak pro studenty ČZU, tak pro potenciální databázové návrháře, kteří hledají inspiraci k tvorbě vlastní databáze, byť se týká jiné problematiky.

Cílů jsem dosáhl studiem a analýzou teoretických poznatků z dostupné literatury vztahující se k tématu bakalářské práce. Povědomí o základech databází a základní terminologii jsem získal díky pracovním zkušenostem a studiu na ČZU.

2.2 Metodika

Metodika řešené problematiky je založena na analýze a studiu dostupných odborných informačních zdrojů. Existující řešení je téměř nemožné identifikovat z důvodu dohody o mlčenlivosti, kterou se v oblasti informačních technologií běžně zaměstnanci zavazují k utajení firemních informací a dat. Z tohoto důvodu lze pouze vycházet ze současných potřeb a předpokladů dané problematiky, řešené v kapitole 4.

Vlastní řešení je realizováno v souladu s datovými požadavky, kterou má databáze podporovat, a to pomocí definice entit, jejich atributů a vazeb mezi entitami.

Na základě syntézy teoretických poznatků, praktických zkušeností a výsledků vlastního řešení jsou formulovány závěry bakalářské práce.

3 Teoretická východiska

3.1 Databáze a její stručná historie

Databáze je soubor vzájemně souvisejících dat, používaný k modelování organizačních struktur nebo organizačních procesů. Na způsobu shromažďování dat nezáleží, mohou být psána i na papír a stále půjde o databázi. V tomto případě se ale bude jednat o počítačový program pro správu dat.^[4]

Základní vývoj databázového softwaru započal v 70. letech, kdy se povedlo pomocí různých metod a algoritmů oddělit data od samotné aplikace. Vzniklo hodně databázových modelů, ovšem v posledních 20 letech šlo převážně o rozvoj relačních systémů, kde se k datům přistupuje pomocí tabulek. Relační databáze (zkráceně RDBMS) byla poprvé představena v roce 1969 jejím zakladatelem E.F.Coddem. První komerční systémy založené na relačním modelu se začaly objevovat na přelomu 70. a 80. let. V současné době se používají relační databáze téměř v každém podniku a vytvořily tak průmysl s obratem v miliardách dolarů ročně.^[3]

3.2 Databázový systém

Databázový systém je výsledkem dlouholeté snahy zjednodušit a zpřístupnit správu dat širší škále lidí, a ne jen zkušeným programátorům, kteří stáli na samotném začátku jeho vývoje.

Databázový systém (DBMS) je občas v české literatuře označován také jako Systém řízení báze dat (SŘBD). Zajišťuje všechny základní služby, které jsou nezbytné pro správu databáze. Těmito službami máme na mysli například současný přístup více uživatelů k datům s ošetřením proti konfliktům, které mohou vznikat aktualizací dat prováděných více uživateli. Další významnou službou je podpora dotazovacího jazyku, který umožňuje získat data z databáze pomocí množiny příkazů. Za zmínku ještě stojí mechanismy pro zálohování databáze, pro zotavení po haváriích, kontrola redundance dat a bezpečnostní mechanismy zabráňující v neoprávněném přístupu k datům.^[2]

Mezi nejrozšířenější softwarové produkty, které lze považovat za databázový systém, můžeme uvést Microsoft Access, DB2, Oracle, Microsoft SQL Server, Sybase, MySQL.

V této práci je cílem navrhnout relační databázový model, který chápat jako architekturu. DBMS je pak podle ní schopen ukládat objekty do databáze a vzájemně je mezi sebou provázat. S tímto modelem přišel E.F.Codd kvůli nedostatkům tehdejších databázových modelů. Založil ho na základě teorie množin a predikátové logice, kdy se snažil vyřešit problémy s redundancí dat a nízkou integritu dat. Relační model byl zaveden jako standard pro popis datových struktur a nad ním byl sestaven dotazovací jazyk relační algebra, který umožňuje formulovat dotazy nad tímto modelem a získávat tak relevantní informace.^[2]

Data jsou v relačním modelu reprezentována pomocí dvourozměrných tabulek jako uspořádané n-tice záznamů a atributů (polí), mezi kterými existují vztahy. Uživatel nepotřebuje znát, kde je daná informace fyzicky uložena. To z důvodu, že každý záznam v tabulce je identifikován polem obsahující unikátní hodnotu a na uspořádání záznamů v databázi nezáleží.

V relačním modelu se rozlišují tři typy vztahů (tj. relací) mezi tabulkami:

1. Jedna k jedné (1:1)
2. Jedna k více (1:N)
3. Více k více (M:N)

Jednotlivé vztahy jsou podrobněji popsány v kapitole 3.3.

K datům lze v relačním modelu přistupovat téměř neomezeným počtem způsobů, jedinou podmínkou je znát vazby mezi tabulkami. Je možné zobrazit i data z tabulek, které jsou v nepřímém vztahu, ale je zapotřebí připojit (joinout) další tabulku, přes kterou se na požadovaná data lze odkázat.^[4]

Výhodou relační databáze je integrita dat, která zajišťuje, že záznamy v tabulkách nejsou duplicitní a detekuje chybějící hodnoty primárních klíčů. Další výhodou spočívá ve fyzické a logické nezávislosti dat na databázové aplikaci. Čili jakékoli uživatelské změny provedené v logickém návrhu, ani fyzické změny na straně poskytovatele nebudou mít vliv na aplikaci. Jako další výhody mohou zmínit snadné získávání dat, konzistenci a přesnost dat.^[2]

3.3 Základní pojmy

Je zapotřebí definovat pojmy, které se vyskytují v oblasti relačních databází. Důvodem je snažší a přesnější vyjádření vlastního procesu návrhu, kde budu používat právě tuto terminologii.

Data

Data jsou veškeré údaje uchovávané v databázi. Zůstávají statická (neměnná) až do doby, než je uživatel nebo nějaký automatický proces změni.

Informace

Informaci lze získat smysluplným zpracováním dat. Ve výsledku je informace srozumitelná a uživatel jí dokáže využít. Narozdíl od dat jsou informace dynamické, protože se dají prezentovat velkým množstvím způsobů a také se mění pokud dojde ke změnám v samotných datech.^[2]

Entita

Entita je libovolná, přesně definovaná množina dat charakterizovaná atributy. Jedná se v podstatě o “objekt” z reálného světa, o kterém se v databázi uchovávají data. Entity bývají v modelech znázorněny jako čtverce reprezentující celou třídu entit. Později v mém návrhu budu mít například entitu Zaměstnanec, která bude vyjadřovat celou kolekci všech zaměstnanců v daném podniku. Jednotliví zaměstnanci pak budou “instancemi” této entity. Rozlišují se ještě externí entity, což jsou entity, o kterých databáze přímo neshromažďuje údaje, ale získává je z externí databáze.^[2]

Atribut

Atribut, občas označovaný také jako “pole”, lze ve fyzickém modelu databáze vnímat jako jednotlivý sloupec tabulky. Atribut určitým způsobem specifikuje danou entitu, ve které se nachází. Jedná se o nejmenší pojmenovanou jednotku dat v databázi, každý atribut by tedy měl být už dále nedělitelný – atomický. Atribut by nikdy neměl obsahovat vypočítané hodnoty, více hodnot toho samého typu a složené hodnoty. Pokud například bych chtěl uchovávat v atributu “jmenoOsoby” jméno i příjmení a aplikace by nevyužívala nikde oddělené údaje, je možné takto atribut ponechat. Metodicky by však

mělo dojít k rozdělení pro případné změny v aplikaci, kdy by bylo zapotřebí použít samostatné jméno nebo příjmení. ^[2]

Ještě je třeba zmínit, že každému atributu musí být přiřazen jedinečný název v rámci své tabulky. To znamená, že více tabulek může obsahovat atribut se stejným názvem, protože při vyhledávání dat poslouží k jejich identifikaci právě název tabulky.

Relace

Relace, neboli vztahy, se vyskytují mezi tabulkami, které spolu logicky souvisí. Umožňují tak zobrazit najednou data z několika tabulek a napomáhají k minimalizaci redundantních (nadbytečných dat). V diagramech jsou relace znázorňovány jako čáry mezi jednotlivými entitami, kdy na každém konci čáry je pomocí speciálních symbolů znázorněna kardinalita vztahu. Ta určuje, kolik instancí jedné entity může být přidruženo k instanci druhé entity.

Vztah lze zřídit prostřednictvím primárních a cizích klíčů, které jsou popsány v pojmech níže. Existují tři druhy relací, každá z nich se využívá pro přidružení záznamů za různých podmínek^[2].

Relace jedna k jedné (1:1)

Pokud je mezi dvojicí tabulek relace 1:1, znamená to, že jeden záznam v první tabulce může být přidružen pouze k jednomu záznamu v druhé tabulce a naopak. V tomto vztahu je vždy jedna tabulka označována jako “rodič” a druhá “potomek”. Vztah 1:1 vznikne zkopírováním primárního klíče z rodičovské tabulky do tabulky potomka. Pouze v tomto případě je možné, aby obě tabulky sdílely stejný primární klíč. V potomkovi tedy bude pole zvolené jako primární klíč zastávat zároveň funkci cizího klíče. ^[1]

Relace jedna k více (1:N)

Relace typu jedna k více existuje mezi dvěma tabulkami, pokud jeden záznam v první tabulce může být přiřazen k jednomu nebo více záznamům v druhé tabulce a zároveň každý záznam v druhé tabulce může být přiřazen pouze jednomu záznamu v první tabulce.

Tabulku na straně “1” je rodič, zatímco tabulka na straně “N” je potomek. Vztah vznikne opět zkopírováním primárního klíče z tabulky rodiče a jeho začleněním do tabulky potomka. Tento vztah je nejčastější typ vztahu, který se obecně vyskytuje v databázi. ^[1]

Relace více k více (M:N)

Relace více k více existuje mezi dvěma tabulkami, pokud jeden záznam v první tabulce může být přiřazen k nule, jednomu nebo více záznamům v druhé tabulce, a naopak. Tento typ relace se musí řešit speciálně pomocí vazební (konceptuální) tabulky.

Vazební tabulka umožňuje spojení záznamů z jedné tabulky se záznamy z druhé tabulky. Dá se říci, že se relace M:N rozdělí na 2 relace typu 1:N a vazební tabulka leží v obou těchto relacích na straně “více” (N). Tabulka se vytvoří nakopírováním primárních klíčů obou původních tabulek. Ty pak dohromady tvoří složený primární klíč a zároveň fungují jako cizí klíče pro obě propojované tabulky.^[1]

Typy účastí ve vztahu

Ve vztazích je důležité definovat pro tabulky typy účastí. Účast může být povinná nebo volitelná. Značení typu účasti je vysvětleno v části “Konceptuální návrh databáze”.

Pokud je účast první tabulky povinná, tak před vložením záznamu do druhé tabulky je potřeba vložit alespoň jeden záznam do první tabulky.

Pokud je účast první tabulky volitelná, tak před vložením záznamu do druhé tabulky není třeba vkládat žádný záznam do první tabulky.^[4]

Tabulka

Tabulky jsou primární strukturou pro ukládání dat v relačním modelu. Každá tabulka reprezentuje jednu specifickou entitu a každý řádek tabulky jednu instanci (tj. výskyt entity). Sloupec tabulky pak reprezentuje atribut této entity.

Každá tabulka musí obsahovat alespoň jedno pole (tj. primární klíč), které identifikuje všechny její záznamy podle své jedinečné hodnoty. Na pořadí záznamů a polí v tabulce nezáleží.^[2]

V relačních databázích se rozlišují datové a validační tabulky. Nejčastějším typem je datová, která slouží k dodávání informací. Validační tabulka (známá také jako číselník nebo vyhledávací tabulka) oproti tomu obsahuje data pro implementaci integrity dat. Ukládá v sobě zpravidla omezenou množinu dat, podle které se poté validuje vkládání dat do datových tabulek. Číselník se obvykle skládá ze dvou polí, kdy první pole slouží jako

primární klíč a druhé uchovává množinu hodnot. Využití číselníků spočívá zejména v aplikačních (business) pravidlech.^[4]

Důležité je dodržovat určité konvence pro pojmenovávání tabulek, aby byl systém přehledný. Zpravidla se používají podstatná jména složená z malých nebo velkých písmen a případná mezera u víceslovného názvu se nahrazuje podtržítkem. Záleží také ale na systému, protože některé nepovolují zápis názvů jinak, než velkými písmeny.

Datový typ

Ke každému atributu, neboli sloupci v tabulce se přiřazuje tzv. datový typ. V dnešní době jsou datové typy definovány v ANSI standardech, které podporuje většina výrobců relačních databázových systémů. Datový typ umožňuje “formátovat” sloupec. Zvolený datový typ specifikuje, jaký typ dat může sloupec obsahovat.

Mezi nejznámější typy patří znakový s pevnou délkou, znakový s proměnlivou délkou, dlouhý textový, celočíselný, desítkový číselný, měna, datum a čas. Každý datový typ je označován zkratkou v závislosti na databázovém systému, kde se označení může lišit. Například v Microsoft Access je celočíselný datový typ označován INTEGER, zatímco v Oracle se používá označení NUMBER.^[4]

Záznam

Jako záznam je označována unikátní instance entity určité tabulky. Skládá se ze všech polí, která daná tabulka obsahuje. K identifikaci každého záznamu se využívá primárního klíče obsahující unikátní hodnotu – zpravidla id. Podle záznamu v jedné tabulce lze určit, jaký je její vztah k záznamu v jiné tabulce.

Klíče

Klíč je speciální druh pole. V závislosti na tom, o jaký klíč se jedná, se určuje role daného pole.

Nejvýznamějším klíčem je primární klíč. Je to pole, které jednoznačně identifikuje každý záznam v tabulce. Pokud je složen ze dvou nebo více polí, nazývá se složený primární klíč. Jeho unikátní hodnoty dokáží identifikovat dané záznamy v celé databázi a pole primárního klíče indentifikuje danou tabulku v rámci celé databáze. Kromě toho pomáhá zřizovat také vztahy mezi tabulkami, protože k propojení tabulek dochází skrze

primární a cizí klíče tabulek. Integrita dat na úrovni tabulky je vynucena právě unikátností hodnot primárního klíče.

Dalším důležitým klíčem je cizí klíč. Využívá se pro zřizování vztahu mezi dvěma tabulkami. Vzniká tak, že se zkopíruje primární klíč první tabulky a vloží se do druhé tabulky, kde chceme zřídit vztah. Název primárního klíče v jedné a cizího klíče ve druhé tabulce nemusí být stejný, ale kvůli přehlednosti a rychlé orientaci v databázi to bývá pravidlem.

Cizí klíč implementuje a zajišťuje integritu dat na úrovni vztahů, protože hodnota cizího klíče musí vždy odpovídat hodnotě primárního klíče, na který se odkazuje. Tím pádem mezi záznamy v obou tabulkách existuje vztah a jsou ošetřeny tzv. "sirotčí" záznamy. Čili pokud existuje záznam v jedné tabulce, musí existovat záznam i v druhé^[2].

Integritní omezení

Omezení nebo také integrita dat, je pravidlo, které určitým způsobem omezuje přípustné datové hodnoty databázového objektu. Objektem je myšlena tabulka nebo sloupec. Integrita dat je jedním z nejdůležitějších aspektů při návrhu databáze. Každé omezení má svůj jedinečný název, přes který se na něj lze odkazovat v chybových zprávách či databázových příkazech, takže uživatel snadno rozpozná o jaké omezení se jedná.^[4]

Existuje několik druhů omezení, základním je omezení primárního klíče (tj. entitní integrita). Toto omezení je implementováno databázovým systémem ve chvíli, kdy je definován primární klíč tabulky. Zaručuje, že v každém řádku bude v poli primárního klíče jiná hodnota a nebude obsahovat hodnotu null. Při vkládání nového řádku (záznamu) systém kontroluje, zda se již hodnota primárního klíče v tabulce nenachází. V případě, že nalezne stejnou hodnotu, systém proces odmítne a ke vložení záznamu nedojde.

Referenční omezení (tj. referenční integrita) je dalším typem omezení, které v databázi zajišťuje relaci mezi tabulkami. Databázový systém kontroluje, zda ke každé hodnotě cizího klíče existuje odpovídající hodnota primárního klíče, a tím zajištěn platný vztah. Ten je důležitý zejména při zadávání, úpravě či mazání dat v jedné z tabulek, aby se změny projevíly také v druhé a nevznikala nesmyslná data. Mohlo by se tak stát, že by v mém návrhu databáze existoval zaměstnanec, který by nebyl zaměstnán v žádné firmě,

pokud bych danou firmu odstranil z tabulky. Většina systémů podporuje tzv. kaskádové odstranění, které umožňuje automaticky odstranit příslušné řádky v dceřiné tabulce, společně s navázaným záznamem v rodičovské tabulce. Ovšem ne ve všech případech je to žádané – například pokud bych odstranil zaměstnance a chtěl ponechat záznamy v dceřinné tabulce o tom, jaká školení zaměstnanec absolvoval.

Doménová integrita zajišťuje, že každá hodnota v daném sloupci má požadovaný typ, rozsah apod. To lze zajistit pomocí jazyka SQL zápisem upraveného příkazu CHECK. Pro složitější omezení se využívají tzv. trigger, neboli spouštěče. Pokud je vyžadováno, aby se zaměstnanec, který ještě nesplnil základní vstupní školení, nemohl zapsat na žádná další školení, trigger by zajistil toto omezení. Spustil by se ve chvíli, kdy by se systém pokusil vložit nový řádek se školením do tabulky. Vyhodnotil by kladně podmínku v triggeru a pokus o vložení by skončilo chybou, která následně zastaví operaci vkládání.^[4]

3.4 Datová normalizace

Normalizace je technika používaná pro vytvoření sady tabulek s minimální redundancí, která podporuje datové požadavky organizace.

Je jedním ze základních předpokladů pro správný logický návrh databáze. Existují určité požadavky na uspořádání dat v tabulkách, kterým se říká normální formy. Výsledkem uplatnění těchto forem v databázi, jsou plně normalizovaná data. Účel normalizace tedy spočívá v odstranění anomálií v datech, které se vyskytují v tabulkách s redundantními (nadbytečnými) daty.

V praxi se využívají převážně první tři normální formy, protože do třetí normální formy lze převést všechny relace a v rozsáhlých databázích je obtížné zajistit dodržení dalších forem. Lze říci, že čím vyšší normální formy je dosaženo, tím je relace kvalitněji navržena. Definice jednotlivých normálních forem a postup, jak jich dosáhnout je popsán v následujících oddílech^[1].

Anomálie vkládání

Redundance je při návrhu struktury databáze nežádoucí tím, že se na více místech opakují stejná data a hrozí situace, kdy se opomene některá z těchto redundantních informací aktualizovat a při vkládání dat dojde k nekonzistenci. Příčinou této anomálie

jsou atributy více entit, které jsou drženy společně v jedné tabulce. Poté nelze vložit pouze částečný záznam do tabulky a je potřeba vyplnit i zbývající atributy hodnotou null.^[4]

Anomálie vymazání

Tato anomálie je opakem anomálie vkládání. Pokud jsou v jedné tabulce drženy atributy více entit a vymaže se záznam jedné konkrétní entity, dojde k nežádoucí ztrátě dat druhé entity. Z tohoto důvodu se provádí proces normalizace, aby došlo k adekvátní dekompozici dat a odstranily se pouze námi požadovaná data.^[4]

Anomálie modifikace (aktualizace)

Anomálie modifikace znamená, že při aktualizaci jediné datové hodnoty se musí aktualizovat více řádků dat. To z důvodu redundance dat, kdy je daná hodnota nadbytečně uložena u více záznamů. Řešením je opět normalizace dat, kdy dojde k separaci atributu do samostatné relace a databázový systém k ní bude přistupovat skrze primární či cizí klíč. Uložena tak bude pouze na jednom místě a při aktualizaci hodnoty nebude docházet k nekonzistenci dat.^[4]

3.4.1 První normální forma (1.NF)

1.NF je jedinou normální formou, která je kriticky důležitá pro vytvoření tabulek v relační databázi. Avšak, aby se předešlo anomáliím aktualizace, autor knihy Mistrovství databáze, T.Connolly, doporučuje používat třetí normální formu.

Relace je v první normální formě, pokud neobsahuje žádné atributy s násobnými hodnotami (atributy musí být atomické). V relaci tak každý průsečík řádku a sloupce musí obsahovat jen jedinou hodnotu. Další problém, který řeší 1.NF, jsou opakované skupiny atributů, díky nimž se vyskytují násobné hodnoty v relaci pro daný atribut.

První normální formu uplatníme na nenormalizovanou relaci přesunutím atributů s násobnými hodnotami a opakovanými skupinami do nové relace, kterou je třeba definovat. Všechny atributy, které se v relaci opakovaly společně, musí být přesunuty do stejné nové relace. Atribut, který v sobě obsahuje více hodnot pak musí být přesunut také do jiné nové relace, nesmí se kombinovat s opakovanou skupinou atributů ve stejné relaci.

Samotný postup pro přesunutí atributů je následující:^[1]

1. Vytvoří se nová relace s odpovídajícím názvem k původní relaci
2. Zkopíruje se primární klíč z původní relace do nové relace. V nové relaci tak bude zastávat funkci cizího klíče, který při tvorbě dotazů umožní propojit zpět oddělené atributy z původní tabulky.
3. Do nové relace se přesune opakovaná skupina nebo atribut s násobnou hodnotou. Přesunuté atributy se tedy budou nacházet pouze v nové relaci, ze staré budou vyjmuty a ze zkopírovaného klíče se stane jedinečný identifikátor, protože bude jednoznačně definovat všechny záznamy přidaných atributů.

3.4.2 Druhá normální forma (2.NF)

Relace je ve 2.NF, pokud je již v 1.NF a hodnoty každého atributu, který není součástí primárního klíče, jsou determinovány všemi hodnotami atributů, které tvoří primární klíč. Tudiž všechny neklíčové atributy musí být funkčně závislé na celém primárním klíči, nikoli pouze části primárního klíče. 2.NF tak ostraňuje částečné závislosti neklíčových atributů.

Převod do 2.NF se týká pouze relací se složenými primárními klíči, tzn. primárními klíči, které jsou tvořeny dvěma a více atributy. Relace v 1.NF, jejichž primární klíč je tvořen jediným sloupcem, jsou automaticky ve 2.NF.

Pro odstranění částečných závislostí se musí odstranit sloupce, které nejsou součástí primárního klíče a které lze určit pomocí pouze části primárního klíče. Tyto sloupce se umístí do nové tabulky společně se zkopírovaným primárním klíčem původní tabulky. Nová tabulka pak bude obsahovat primární klíč tvořený jediným atributem, což je předpoklad pro 2.NF. Původní tabulka bude rovněž ve 2.NF, protože se v ní již nebudou nacházet žádné neklíčové atributy závislé pouze na části primárního klíče^[1].

3.4.3 Třetí normální forma (3.NF)

Relace je ve 3.NF, pokud je již v 1.NF a 2.NF a všechny hodnoty v neklíčových attributech jsou determinovány pouze atributy primárního klíče a žádnými dalšími atributy.

Účelem 3.NF je odstranit tranzitivní závislosti mezi atributy. Tranzitivně závislý atribut je takový atribut, který není definován jako primární klíč a zároveň je závislý na

jiném neklíčovém atributu. Výsledkem má být relace, kde jsou všechny neklíčové atributy závislé pouze na primárním klíči.

Převod z 2.NF do 3.NF spočívá v přesunutí všech tranzitivně závislých atributů do jiných relací, kde jsou závislé pouze na svém primárním klíči. Atribut, na kterém byly přesouvané atributy tranzitivně závislé, zůstává v původní relaci a zároveň se stává cizím klíčem k novým relacím, kde jeho kopie zastává funkci primárního klíče. Všechny relace se tak nachází ve 3.NF, protože jsou závislé pouze na svém primárním klíči^[1].

3.4.4 Boyce-Coddova normální forma (BCNF)

BCNF říká, že relace musí být ve 3.NF a nesmí existovat jiné funkční závislosti, nežli sekundárních atributů (tj. neklíčové atributy) na primárních (tj. atributy primárního klíče). Oproti 3.NF zde nesmí existovat závislost mezi primárními atributy, ani závislost primárního atributu na sekundárním atributu. Z důvodu takto přísných kritérií nemusí BCNF existovat pro všechny relace.

Řešením pro relaci, která nesplňuje požadavky BCNF, je vyčlenění nežádoucí vazby do samostatné tabulky, podobně jako u 3.NF.^[4]

3.5 Metodologie návrhu databáze

Metodologií návrhu se rozumí „*strukturovaný přístup používající procedury, techniky, nástroje a dokumentaci s cílem podpořit a usnadnit proces návrhu.*“^[1]

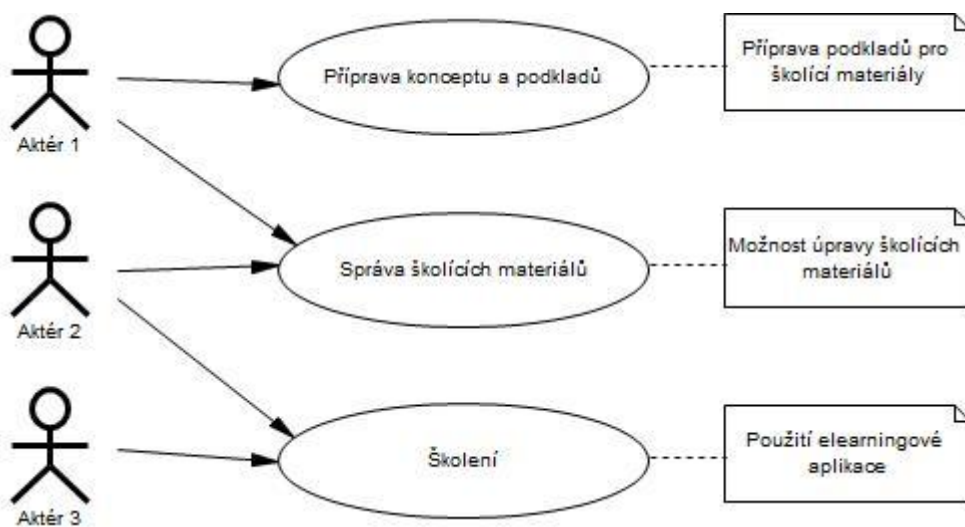
Je tvořena třemi hlavními stádii, a to konceptuálním, logickým a fyzickým návrhem. Všechny tyto fáze jsou nedílnou součástí životního cyklu databáze.

3.5.1 Konceptuální návrh databáze

Jedná se o proces vytvoření konceptuálního modelu dat na základě dat používaných organizací bez jakýchkoli úvah o fyzické implementaci. V této fázi se neberou v potaz podrobnosti, jako například relační datový model. Cílem konceptuálního návrhu je identifikovat důležité entity a jejich relace, které poté slouží jako zdroj informací pro logický model dat.^[1]

K určení entit se využívá funkční specifikace, která definuje uživatelské požadavky na databázi. Právě sběr a analýza požadavků je předběžnou fází návrhu databáze. Množství sebraných dat závisí na rozsáhlosti požadavků firmy a jejich pracovních postupů.


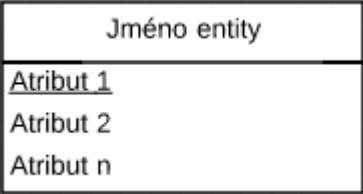
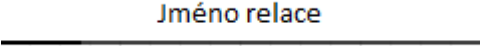
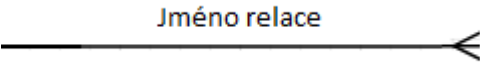
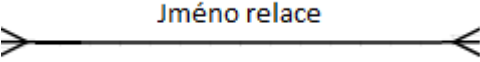
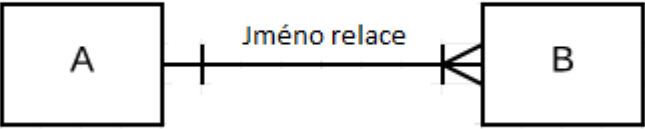
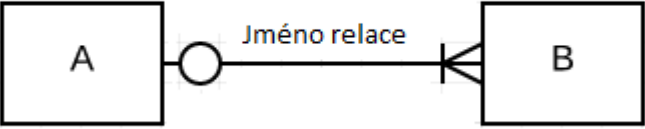
Další dokumentací využívanou při návrhu databáze, je diagram případů užití (v angličtině *Use case diagram*), který patří do rodiny UML diagramů. Ten zobrazuje chování systému z pohledu uživatele. Z diagramu tak vyplývá, co má systém umět, ale přímo nespecifikuje, jak toho má být dosaženo. Právě tohoto typu diagramu využijí při návrhu řešení pro zaznamenání mnou kladených požadavků, kterým se budu věnovat v kapitole 4.

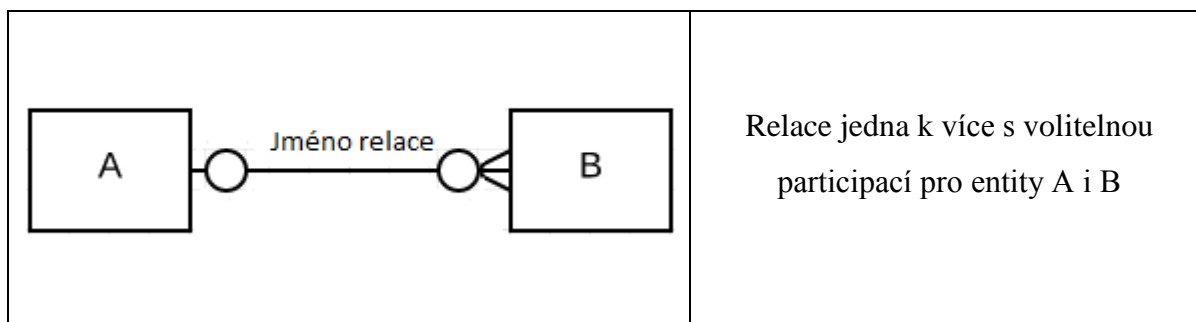


Obrázek 1 - Use case diagram - ukázka

Diagram je tvořen jednotlivými aktéry, zastávající v aplikaci určité funkce, které jim umožňují provádět zvolené úkony jako je například správa školicích materiálů. Těmto úkonům se říká případ užití (use case). Pro konceptuální návrh lze z tohoto diagramu částečně vyvodit, jaké vztahy musí vzniknout mezi jednotlivými entitami. Jedná se samozřejmě o specifické grafické zpracování požadavků, čili ze samotné funkční specifikace bych došel ke stejnému závěru. Toto je pouze přehlednější forma, jak zjištěné informace zobrazit. K zakreslení use case diagramů bylo využito nástroje PowerDesigner.

Výsledkem konceptuálního návrhu je Entity-relationship model (dále jen ERM), který umožňuje snazší pochopení uživatelských požadavků na databázový systém. K jeho vytvoření je třeba využít jednu ze standardizovaných notací. Nejvíce rozšířenou v podnikové sféře je notace vraních stop, ze které při návrhu využijí následující prvky^[1]:

Symbol	Význam
	Entita
	Seznam atributů se uvádí uvnitř entity. Podtržený atribut značí primární klíč.
	Relace jedna k jedné
	Relace jedna k více
	Relace více k více
	Relace jedna k více (1:M) s povinnou participací pro entity A i B
	Relace jedna k více (1:M) s volitelnou participací pro entitu A a povinnou participací pro entitu B



Tabulka 1 - Notace vraních stop pro ER modelování^[1]

Notace obsahuje ještě škálu dalších symbolů, které však své využití nacházejí zpravidla u rozsáhlejších modelů. V mém modelu neplánuji zavádět žádné rekurzivní relace, ternární relace či nadtržidy. K zakreslení symbolů bylo využito online nástroje Lucidchart^[5].

Při vytváření ER modelu je nezbytné dodržet následující kroky^[1]:

1. Identifikace entit
2. Identifikace relací
3. Identifikace a spojení atributů s entitami
4. Určení domén atributů
5. Určení atributů, které budou zastávat funkci primárního nebo cizího klíče
6. Kontrola redundance modelu

Jednotlivé kroky jsou probírány v samotném řešení této práce v souvislosti s kladenými požadavky na databázi.

3.5.2 Logický návrh databáze

Jedná se o proces vytvoření logického modelu dat, který je nezávislý na DBMS a jiných fyzických implementacích. Na základě ER modelu se vytvoří tabulky, jejichž struktura se zkontroluje pomocí normalizace. Struktura tabulek vyplývá přímo z ER modelu či jiné podpůrné dokumentace. V této fázi návrhu se rovněž uplatňují integritní omezení.

Kontrola tabulek pomocí pravidel normalizace je nutností. Je třeba zjistit, zda jsou všechny tabulky alespoň ve 3.NF. Pokud tomu tak není, musí se rekonstruovat buď tabulky nebo ER model, ze kterých struktury tabulek vyplývají.

Jak už bylo zmíněno, kontrola integritních omezení je součástí logického návrhu databáze. Je potřeba definovat všechna omezení vyplývající z požadavků uživatele, dokonce i ta, které nebude možné v DBMS implementovat. Musí se tedy určit, která data jsou požadovaná, definovat množiny přípustných hodnot pro atributy, zajistit integritu entit a referenční integritu.

Z výše uvedeného tak vyplývají pro logický návrh tyto úkony^[1]:

1. Vytvoření tabulek
2. Kontrola tabulek pomocí normalizace
3. Kontrola integritních omezení

3.5.3 Fyzický návrh databáze

Fyzický návrh spočívá v implementaci normalizovaných relací, které byly definovány v logickém návrhu, do prostředí konkrétního DBMS. Aplikací příkazů jazyka SQL se fyzicky vytvoří objekty definované v logickém modelu dat. Součástí této fáze je také návrh uživatelských pohledů, bezpečnostních mechanismů či volba organizace souborů a indexů pro dosažení přijatelné výkonnosti databáze. Fyzický návrh není však předmětem této práce.^[4]

4 Zmapování současného stavu a definice požadavků

4.1 Současný stav

Vzdělávání ve firmách se praktikuje již delší dobu a finanční prostředky investované k její realizaci každoročně rostou. Trendem není pouze e-learning, v hledáčku zůstávají také klasické prezenční kurzy buď v sídle společnosti nebo přímo ve specializovaných školicích centrech. Podle průzkumu ING pojišťovny a Svazu průmyslu a dopravy v roce 2014 nabízelo vzdělání pro své zaměstnance 81 procent českých firem. Samotné kurzy pak byly k dispozici zhruba třetině zaměstnanců. Důvodem zvyšování investic do vzdělání bylo v minulých letech způsobeno především zlepšující se ekonomickou situací v České republice. Podle průzkumu Benchmarking 2013 společnosti Top vision, roste poptávka firem po rozsáhlejší integraci e-learningu do firemního vzdělávání z důvodu úspory nákladů^[8].

V současné situaci existuje silná potřeba zavádět informační systémy pro zvyšování kvalifikace zaměstnanců a to z několika důvodů^[7]:

- Stále vyšší kladené nároky na zaměstnance. Podíl na tom má společenský vývoj a rozvoj informačních technologií, kvůli kterým si musí zaměstnanec rozšiřovat své odborné vzdělání a kvalifikaci.
- Zaměstnanci musí být „použitelní“ v širším oboru, aby byli schopni zastávat pracovní činnosti svých kolegů v krizových situacích.
- Kvůli globalizaci jsou zaměstnanci často nuceni zvládnout komunikaci s mezinárodními firmami v různých jazycích, včetně osvojení si jejich kultury a zvyků.
- Snižování nákladů.
- Péče o zaměstnance – konkrétně rozvoj a vzdělání zaměstnanců – zvyšuje reputaci firmy, což se ve výsledku může projevit ve vyšším počtu kandidátů ucházejících se o pracovní pozici ve firmě.
- Konkurenceschopnost. Pravidelné vzdělávání zaměstnanců je nezbytné pro získání konkurenční výhody ve smyslu zvýšení výkonů zaměstnanců.

4.2 Přínosy informačního systému pro vzdělávání zaměstnanců

Pro efektivní vzdělávání je informační systém významným přínosem, obzvláště pokud podporuje kromě správy prezenčních kurzů také e-learningové kurzy. Firma pak může disponovat vyšší flexibilitou zaměstnanců, zvýšenými výkony, lepší úrovní služeb, snížením fluktuace zaměstnanců a zejména snížením nákladů z dlouhodobé perspektivy.

Důležitým aspektem pro firmu jsou rovněž reporty, které lze získat z databáze pomocí dotazů. Vedoucí pracovníci pak mají přehled o svých zaměstnancích, jejich potřebách či naopak nedostatcích.

Firemní vzdělávání je přínosné i z pohledu samotného zaměstnance. Zaměstnanec díky vzdělávání zvyšuje svoji vlastní cenu na trhu práce a umožňuje mu tak nalézt snáze uplatnění v jeho oboru. Získané znalosti a dovednosti jsou v dnešní době velmi žádané a náležitě odměněny. Rozšiřování vzdělání má tedy přímou spojitost s uplatněním zaměstnance na trhu práce. Mimo to má také velký vliv na kariérní růst, který je úzce spjat se mzdovým ohodnocením^[7].

Dalším přínosem pro zaměstnance kromě uplatnění a kariérního růstu, jsou náklady vynaložené na vzdělání. Zpravidla ho firma poskytuje bezplatně, nebo dotuje jeho část. I pokud by měl část nákladů na kurzy krýt zaměstnanec, v jeho vlastním zájmu je to zajímavá nabídka, která ho přiměje se kurzů pravidelně účastnit. Kurzy, které musí částečně hradit zaměstnanec, bývají často zakončovány s možností získat certifikát o úspěšném absolvování kurzu, které opět zvyšují cenu zaměstnance.

4.3 Přístup firem ke zvyšování kvalifikace

Ne všechny firmy si mohou dovolit investovat do informačního systému, jedná se zejména o menší firmy, které firemnímu vzdělávání nevěnují příliš velkou pozornost právě z důvodu malého počtu zaměstnanců. Omezují se tak pouze na školení stanovená zákonem.

K dosažení nějakých výsledků, viditelných změn u svých zaměstnanců, musí docházet k systematickému vzdělávání. Nevzdělávat se pouze podle momentální potřeby, ale ideálně vedoucí pracovníci či lektori by měli vymezit určitý „studijní plán“ a zaměstnanci by měli mít pak možnost poskytnout zpětnou vazbu, podle které by docházelo k úpravě tohoto plánu. Právě chybějící zpětná vazba od zaměstnanců je často příčinou

neefektivního vzdělávání. Informační systém by měl mít navržen svůj datový model tak, aby měl každý zaměstnanec možnost k jednotlivým kurzům či testům dát zpětnou vazbu a to ideálně anonymně, z důvodu pravdivosti údajů ^[7] .

Nespokojenost zaměstnanců s osobním rozvojem ve firmě může mít negativní vliv na jejich pracovní morálku a vyústit ve stagnaci nebo dokonce výpověď zaměstnance. Proto je potřeba systematicky zjišťovat potřeby zaměstnanců a připravovat odpovídající kurzy, které pro ně budou výzvou a impulsem k dalšímu vzdělávání. Podle statistiky institutu National Research Business Institute z roku 2013, téměř 23 procent zaměstnanců ukončilo pracovní poměr právě z důvodu nedostatku příležitostí k osobnímu rozvoji^[9].

4.4 Způsob definování požadavků

Všechny požadavky by v reálném prostředí měly přirozeně vyplynout z různých způsobů zjišťování, jako jsou dotazníky, rozhovory se zaměstnanci či pozorování organizace a jejího provozu. Jelikož se však zpravidla jedná o citlivé interní informace, které nejsou veřejně dostupné, musím vycházet z obecných předpokladů funkčnosti pro uživatele a z nejnovějších trendů společnosti.

Jedním z takovýchto trendů je právě E-learning, který se v posledních letech značně rozšířil a stal se určitým standardem ve firemním vzdělávání. Je to metoda vzdělávání formou elektronické "učebnice", u které není nutná prezenční účast a přímá komunikace s lektorem. Právě k této metodě se bude vztahovat podstatná část mých požadavků.

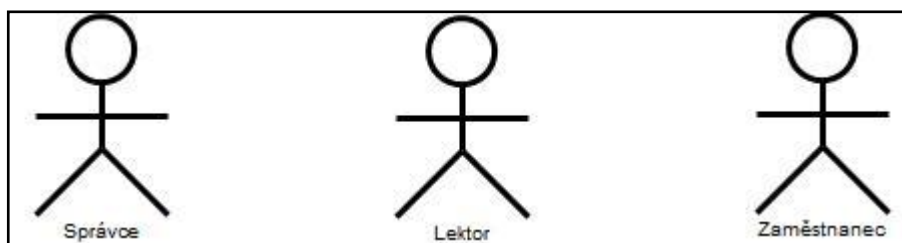
4.5 Typy a charakteristiky uživatelů

Databáze bude podporovat celkem tři typy uživatelů. Každý uživatel systému bude mít jednu z definovaných rolí a s tím i spojené funkce, které může v rámci systému vykonávat. Je to pouze souhrnný popis jejich rolí, je nutné definovat konkrétní dílčí požadavky a ideálně je znázornit formou Use case diagramů.

Role jsou definovány v tabulce 2.

Tabulka 2 – Role uživatelů a jejich popis

Aktér	Popis
Správce	Role pro osobu, která má zodpovědnost za: <ul style="list-style-type: none">- Správu lektorů- Správu školících společností- Správu zaměstnanců
Lektor	Role pro osobu, která má na starosti: <ul style="list-style-type: none">- Přípravu konceptu a podkladů pro školení.- Správu kurzů.
Zaměstnanec	Role pro osobu, která: <ul style="list-style-type: none">- Účastní se kurzů.- Poskytuje zpětnou vazbu k jednotlivým běhům kurzů.



Obrázek 2 - Znázornění uživatelů v Use case diagramech

Tyto tři role jsou definované pro základní, obecný model. Každá organizace se liší svojí procesní strukturou a vyžaduje odlišné role. Mojí snahou je nyní zahrnout do modelu základní funkčnosti pro chod systému školení, aby bylo možné model v budoucnu rozšířit podle požadavků konkrétního podniku.

4.6 Základní požadavky na systém

Cílovým požadavkem je umožnit přihlašování ke kurzům. Kromě prezenčních kurzů je ale potřeba také umožnit přihlašování na e-learningové (dále jen „online“) kurzy. Jsou tedy celkem vyžadovány dvě formy kurzů - prezenční a online.

Lze předpokládat, že podnik bude chtít připravovat pro své zaměstnance různé druhy kurzů, které budou běžet v různých časech. Proto podnik může navazovat vztahy s více školicími společnostmi, jejichž lektoři pro něj připraví sadu kurzů, na které se poté mohou zaměstnanci přihlašovat. Některé kurzy budou muset zaměstnanci po uplynutí určitého období opakovat z legislativních důvodů. Například pro splnění povinnosti zaměstnavatele vyplývající ze Zákoníku práce, musí zaměstnanec úspěšně absolvovat školení BOZP (Bezpečnost a ochrana zdraví při práci), které se musí opakovat nejméně jednou za dva roky. Z tohoto důvodu je nutné pro každý kurz uchovávat informaci, kdy jej daný uživatel splnil. Zároveň to znamená, že každý uživatel může exekuvovat stejný test vícekrát a každý test může být spuštěn více uživateli.

Pro kurzy je potřeba uchovávat školící materiál, který se bude vázat k jednomu nebo více kurzům. Z materiálů mohou uživatelé získat potřebné informace k úspěšnému splnění kurzu. Zaměstnavatele (manažery) budou pak zajisté zajímat výsledky kurzů za jednotlivé zaměstnance. Školící materiály připravuje lektor, který musí mít možnost dodávat materiál průběžně (z důvodu rozsáhlosti některých testů) a je potřeba rozlišovat, zda je materiál dokončený nebo rozpracovaný. Rozpracovaný stav je také potřeba rozlišovat u jednotlivých instancí kurzu, aby se mohl uživatel k delšímu kurzu vrátit. Pro všechny účastníky je nutné evidovat základní personální údaje. Pro přehlednost kurzů je vhodné zavést kategorie kurzů. Kvůli univerzálnosti je požadováno zvolit online testy stylem výběru jedné nebo více správných odpovědí ze sady.

Pokud nelze z kladených požadavků jasně určit entity, atributy či kardinalitu relací, pak záleží na vlastním uvážení, aby dané prvky byly relevantně zavedeny a plnily svůj účel.

4.7 Dílčí požadavky na systém

V základních požadavcích bylo nastíněno hned několik věcí, se kterými musí databáze počítat. Nyní lze definovat jednotlivé dílčí požadavky, které jsou specifitější a lze na základě nich vytvořit Use case diagramy.

4.7.1 Formy školení

Školení mohou probíhat buď interně, tedy interními lektory pracující přímo pro firmu, nebo externě prostřednictvím externích lektorů, pracujících pro školící společnost.

Z požadavku vyplývají následující potřeby:

- Rozlišovat zda se jedná o interního nebo externího lektora.
- Evidovat základní údaje o lektorech.
- Evidovat základní údaje o školících společnostech.
- Školící společnost může dodávat více lektorů, ale dodaný lektor pracuje právě pro jednu školící společnost.

4.7.2 Struktura online testů

Je zvolen typ testu, kdy dochází k výběru jedné nebo více správných odpovědí ze sady již nadefinovaných odpovědí, které přísluší k dané otázce. Tímto způsobem lze zkonstruovat jakýkoli test a vyhodnocení testu není náročné na implementaci. Defaultně budou zavedeny čtyři odpovědi, ze kterých si uživatel bude moci vybrat, atributy dané entity však lze snadno rozšířit o více možných odpovědí. Některé testy mohou být bodované a časově omezené.

Vyplývající potřeby na datový model:

- Evidovat čtyři odpovědi pro každou otázku.
- Umožnit časové omezení testu.
- Umožnit zadat bodové hodnocení ke každé otázce.
- Evidovat, zda je test aktivní či nikoliv. Test může být v rozpracovaném či přerušném stavu.
- Každá zvolená odpověď uživatelem náleží právě té jedné instanci otázky na kterou uživatel odpovídal.

4.7.3 Struktura kurzů

Dá se očekávat, že kurzů bude větší počet a pro lepší orientaci mezi kurzy je tak třeba zavést kategorie kurzů, do nichž se budou kurzy řadit. Kurzy mohou být zpoplatněné a nepřístupné v různých časových intervalech či nepřístupné z důvodu nedostatečného oprávnění. Jednotlivé kurzy mohou být vypsány vícekrát v různých lokalitách a časech. Místo konání kurzu je důležité pouze pro prezenční kurzy, protože online kurzy se vykonávají v místě působiště uživatele. Některé online kurzy mohou mít definovanou povinnost splnit určitý bodový limit. Ke každému absolvovanému kurzu musí uživatel dát zpětnou vazbu ve smyslu několika vět týkajících se kurzu. Jednotlivé instance kurzů pak mají určitou kapacitu obsazenosti.

Z požadavku vyplývají následující potřeby:

- Evidence kategorií kurzů.
- Evidence kurzů, kdy každý kurz spadá právě do jedné kategorie kurzu.
- Rozlišení prezenčních kurzů.
- Povinnost evidovat minimální kapacitu, maximální kapacitu a místo konání pro prezenční kurzy.
- Evidence základních údajů o kurzech, jako název, popis, datum apod.
- Mít možnost zadat cenu ke kurzu.
- Možnost opakovat stejný kurz.
- Povinnost zpětné vazby po absolvování kurzu.
- Zavedení nepovinného minimálního počtu bodů pro online kurzy, kterých by musel uživatel dosáhnout pro úspěšné splnění kurzu.

4.7.4 Práva uživatelů

V kapitole 4.5 - *Typy a charakteristiky uživatelů* byly definovány 3 uživatelské role, kdy každá z nich vykonává různé funkce.

Z kapitoly vyplývají tyto potřeby:

- Pro všechny role evidovat základní uživatelské údaje, jako je jméno, příjmení, uživatelské jméno, kontakt a podobně.
- Evidovat, zda je uživatel aktivní – zda-li může pracovat se systémem.
- Určit pro každého uživatele uživatele jednu z rolí.
- Umožnit uživatelům spouštět testy
- Evidovat, zda se jedná o interního nebo externího zaměstnance.

- Možnost vyhledat uživatele s jakoukoli rolí podle unikátního ID.

4.7.5 Práce s dokumenty

Je žádoucí uchovávat ke kurzům školící materiály. Každý kurz může mít materiálů více. O jejich správu se stará lektor, který materiál vytvořil. Lektoři si nemohou navzájem materiály upravovat.

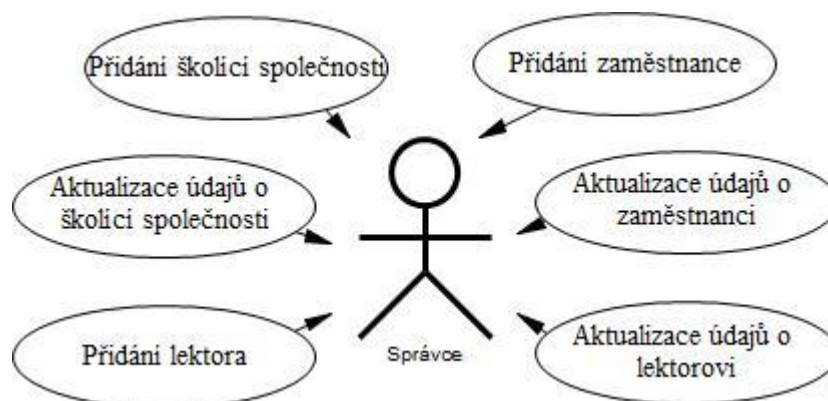
Z požadavku vyplývají tyto potřeby:

- Evidovat základní údaje o školících materiálech (název, datum vložení apod.)
- Umět rozlišit autora materiálu
- Možnost kurzu disponovat více materiály
- Rozlišovat rozpracovaný a aktivní stav materiálu

4.8 Pohledy uživatelů

Uživatelský pohled v této práci slouží k definování, co se vyžaduje od systému z pohledu konkrétní role. Každá role, kterou jsem charakterizoval v kapitole 4.5, bude vyžadovat částečně odlišná data pro svoji práci. Pomocí Use case diagramu (tj. Diagram případů užití) vytvořím uživatelské pohledy pro jednotlivé uživatele, aby bylo jasné, jaká osoba s určitou rolí má provádět jaké operace. Některé případy užití jsou společné pro více rolí.

V diagramu případů užití je každý případ užití spojen „komunikační“ čarou s jedním nebo více aktéry, kteří znázorňují uživatelský pohled.



Obrázek 3 - Use case diagram pro uživatelský pohled Správce

Tabulka 3 - Popis use case diagramu pro uživatelský pohled Správce

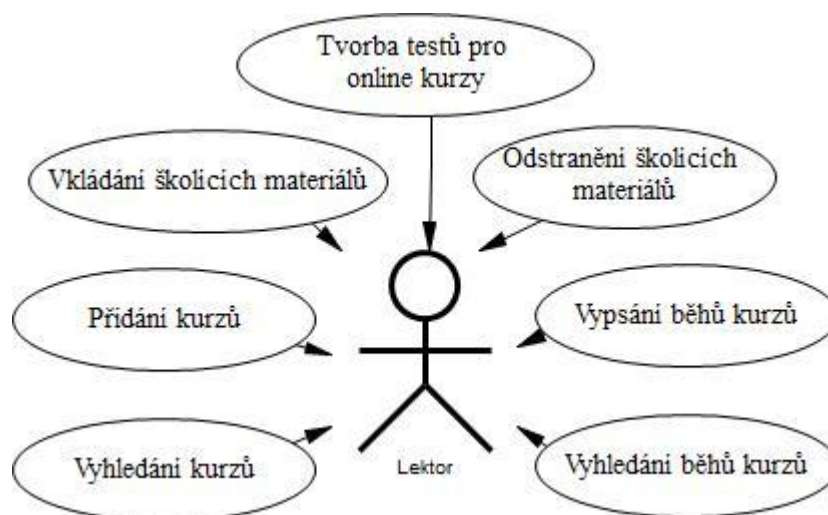
Případ užití	Popis
1. Přidání školící společnosti	Vytvoření záznamu pro novou školící společnost. Každá společnost je jedinečně určena pomocí identifikátoru společnosti.
2. Aktualizace údajů o školící společnosti	Pro aktualizaci údajů o školící společnosti.
3. Přidání zaměstnance	Vytvoření záznamu pro nového zaměstnance. Každý zaměstnanec je jedinečně určen pomocí čísla zaměstnance.
4. Aktualizace údajů o zaměstnanci	Pro aktualizaci údajů o zaměstnanci.
5. Přidání lektora	Vytvoření záznamu pro nového lektora. Každý lektor je jedinečně určen identifikátorem lektora a každý lektor pracuje pro jednu školící společnost.
6. Aktualizace údajů o lektorovi	Pro aktualizaci údajů o lektorovi.



Obrázek 4 - Use case diagram pro uživatelský pohled Zaměstnanec

Tabulka 4 - Popis use case diagramu pro uživatelský pohled Zaměstnanec

Případ užití	Popis
1. Přihlášení k běhu kurzu	Jakmile zaměstnanec vyhledá požadovaný kurz, bude se moci přihlásit k jednomu z vypsaných běhů tohoto kurzu.
2. Odhlášení z instance kurzu	Pokud je zaměstnanec přihlášen k běhu kurzu, bude se moci do určitého času od zahájení instance z ní odhlásit.
3. Exekuce instancí online kurzu	Zaměstnanec spustí a vypracuje online test.
4. Vyhledávání kurzů	Zaměstnanec vyhledá kurz podle zadaných kritérií.
5. Vyhledávání instancí kurzů	Zaměstnanec vyhledá běh kurzu podle zadaných kritérií.
6. Aktualizace údajů o zaměstnanci	Pro aktualizaci údajů o zaměstnanci.



Obrázek 5 - Use case diagram pro uživatelský pohled Lektor

Tabulka 5 - Popis use case diagramu pro uživatelský pohled Lektor

Případ užití	Popis
1. Tvorba testů pro online kurzy	Přidání testu pro online kurz.
2. Vkládání školících materiálů	Lektor vloží ke kurzu školící materiál.
3. Odstranění školících materiálů	Lektor odstraní školící materiál z kurzu.
4. Přidání kurzů	Vytvoření záznamu pro nový kurz. Každý kurz je jedinečně určen pomocí identifikátoru kurzu.
5. Vypsání běhů kurzů	Vytvoření záznamu pro nový běh kurzu. Každý běh je jedinečně určen pomocí identifikátoru běhu kurzu.
6. Vyhledávání kurzů	Lektor vyhledá kurz podle zadaných kritérií.
7. Vyhledávání běhů kurzů	Lektor vyhledá běh kurzu podle zadaných kritérií.

5 Řešení konceptuálního návrhu

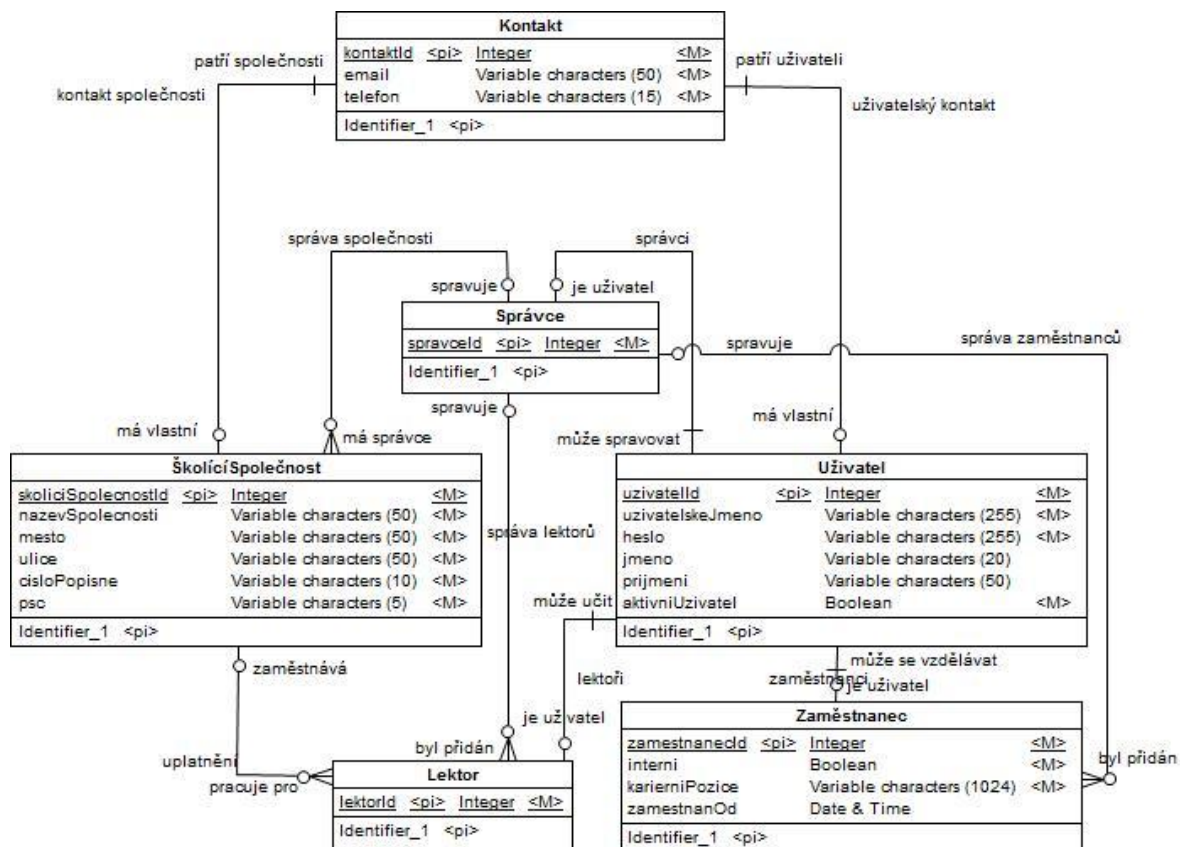
Výsledný konceptuální model identifikuje jednotlivé entity a jejich atributy takovým způsobem, aby zohledňoval současnou definici požadavků na aplikaci.

Součástí této kapitoly je kromě sestavení finálního modelu především popsání všech entit, atributů a vztahů mezi entitami. Pro lepší orientaci v modelu jsem ho pomyslně rozdělil na tři logické celky, které jsem pomocí nástroje PowerDesigner barevně odlišil ve finálním datovém modelu. Jedná se o celky *Správa osob*, *Správa kurzů* a *Správa testů*, které, jak už z názvu vyplývá, zahrnují pro ně specifickou sadu entit.

5.1 Správa osob

Tento celek má za účel zajistit evidenci základních údajů o osobách, které byly zmíněny v požadavcích. Různé typy osob musí v aplikaci disponovat jinou sadou oprávnění a je tak třeba rozlišovat uživatelské role (podrobněji jsou popsány v kapitole 4.5).

Předpokládá se, že výsledná aplikace bude muset přístupová práva a nastavení rolí řešit ne pouze aplikačně, ale také databázově. Proto jsem definoval role jako samostatné entity, které by potenciálně mohly v budoucnu držet specifická nastavení oprávnění. Pokud by to nebylo třeba, stačilo by přidat atribut `typUzivatele` do společné entity `Uzivatel`, který by se následně plnil enumerací obsahující typy uživatele.



Obrázek 6 - Část konceptuálního modelu - Správa osob

5.1.1 Entity a jejich atributy

V této části jsou vypsány všechny entity týkající se správy osob. Pro každou entitu jsou pak v tabulce znázorněny všechny atributy s formou popisu:

Název atributu – Datový typ – Informace k atributu.

UŽIVATEL

Tabulka 6 - Atributy entity Uživatel

Název atributu	Datový typ	Informace k atributu
uzivatelskeId	Integer	Identifikátor entity
uzivatelskeJmeno	Varchar(255)	Uživatelské jméno
heslo	Varchar(255)	Uživatelské heslo
jmeno	Varchar(20)	Jméno uživatele
prijmeni	Varchar(50)	Příjmení uživatele
aktivniUzivatel	Boolean	Říká, zda je uživatel aktivní

ZAMĚSTNANEC

Tabulka 7 - Atributy entity Zaměstnanec

Název atributu	Datový typ	Informace k atributu
zamestnanecId	Integer	Identifikátor entity
interni	Boolean	Určuje, zda se jedná o interního zaměstnance
karierniPozice	Varchar(50)	Specifikuje kariérní pozici zaměstnance.
zamestnanOd	Date&Time	Určuje od kdy je uživatel zaměstnancem firmy

SPRÁVCE

Tabulka 8 - Atributy entity Správce

Název atributu	Datový typ	Informace k atributu
spravceId	Integer	Identifikátor entity

LEKTOR

Tabulka 9 - Atributy entity Lektor

Název atributu	Datový typ	Informace k atributu
lektorId	Integer	Identifikátor entity
interni	Boolean	Určuje, zda se jedná o interního lektora

ŠKOLÍCÍ SPOLEČNOST

Tabulka 10 - Atributy entity ŠkolícíSpolečnost

Název atributu	Datový typ	Informace k atributu
skoliciSpolecnostId	Integer	Identifikátor entity
nazevSpolecnosti	Varchar(50)	Název školící společnosti
mesto	Varchar(50)	Atributy identifikující zázemí školící společnosti.
ulice	Varchar(50)	
cisloPopisne	Varchar(10)	
psc	Varchar(5)	

KONTAKT

Tabulka 11 - Atributy entity Kontakt

Název atributu	Datový typ	Informace k atributu
kontaktId	Integer	Identifikátor entity
email	Varchar(50)	Kontaktní údaje osoby
telefon	Varchar(15)	

5.1.2 Vztahy mezi entitami

V této části jsou zmíněny vztahy mezi entitami, které zasahují do celku správy osob. Pro každý vztah jsou definovány dvě entity, typ vztahu a odůvodnění, proč je zvolen daný typ relace.

UŽIVATEL - ZAMĚSTNANEC

Zaměstnanec musí být přiřazen právě jednomu Uživateli a Uživatel může být přiřazen právě jednomu zaměstnanci.

Zvolená kardinalita vztahu je tedy 1:1. Není totiž možné, aby zaměstnanec existoval pod více uživateli, a také není možné, aby jeden uživatel zastával funkci více zaměstnanců.

UŽIVATEL – LEKTOR

Lektor musí být přiřazen právě jednomu Uživateli a Uživatel může být přiřazen právě jednomu Lektorovi. Jedná se o stejný typ vztahu jako Uživatel – Zaměstnanec, kardinalita je 1:1 ze stejných důvodů.

UŽIVATEL – SPRÁVCE

Správce musí být přiřazen právě jednomu Uživateli a Uživatel může být přiřazen právě jednomu Správci. Jedná se o stejný typ vztahu jako Uživatel – Zaměstnanec, kardinalita je 1:1 ze stejných důvodů.

UŽIVATEL – KONTAKT – ŠKOLÍCÍ SPOLEČNOST

Uživatel musí mít přiřazen právě jeden kontakt. Školící společnost musí mít přiřazen právě jeden kontakt. Kontakt musí být přiřazen buď uživateli, nebo školící společnosti.

Je zde nastavena kardinalita 1:1. Z důvodu normalizace byl vytržen kontakt z entit Uživatel a Školící Společnost a byly vytvořeny vztahy právě s těmito entitami. Kontakt je povinný údaj, proto musí vždy být přiřazen záznamu jedné z entit a naopak. Šlo by také uvažovat o kardinalitě 1:M, kdy by uživatel nebo školící společnost mohli mít přiřazeno

více kontaktů (například kvůli síti poboček školící společnosti). V tomto modelu se ovšem vyžaduje pouze jeden záznam kontaktu.

SPRÁVCE – ŠKOLÍCÍ SPOLEČNOST

Správce může mít přiřazeno více školících společností. Školící společnost však může mít pouze jednoho správce.

Kardinalita je u tohoto vztahu 1:M. Logicky správce musí mít možnost spravovat více záznamů školících společností, jinak by musel mít podnik stejný počet správců jako je počet školících společností. Jednalo by se tak o velmi neefektivní způsob správy evidence.

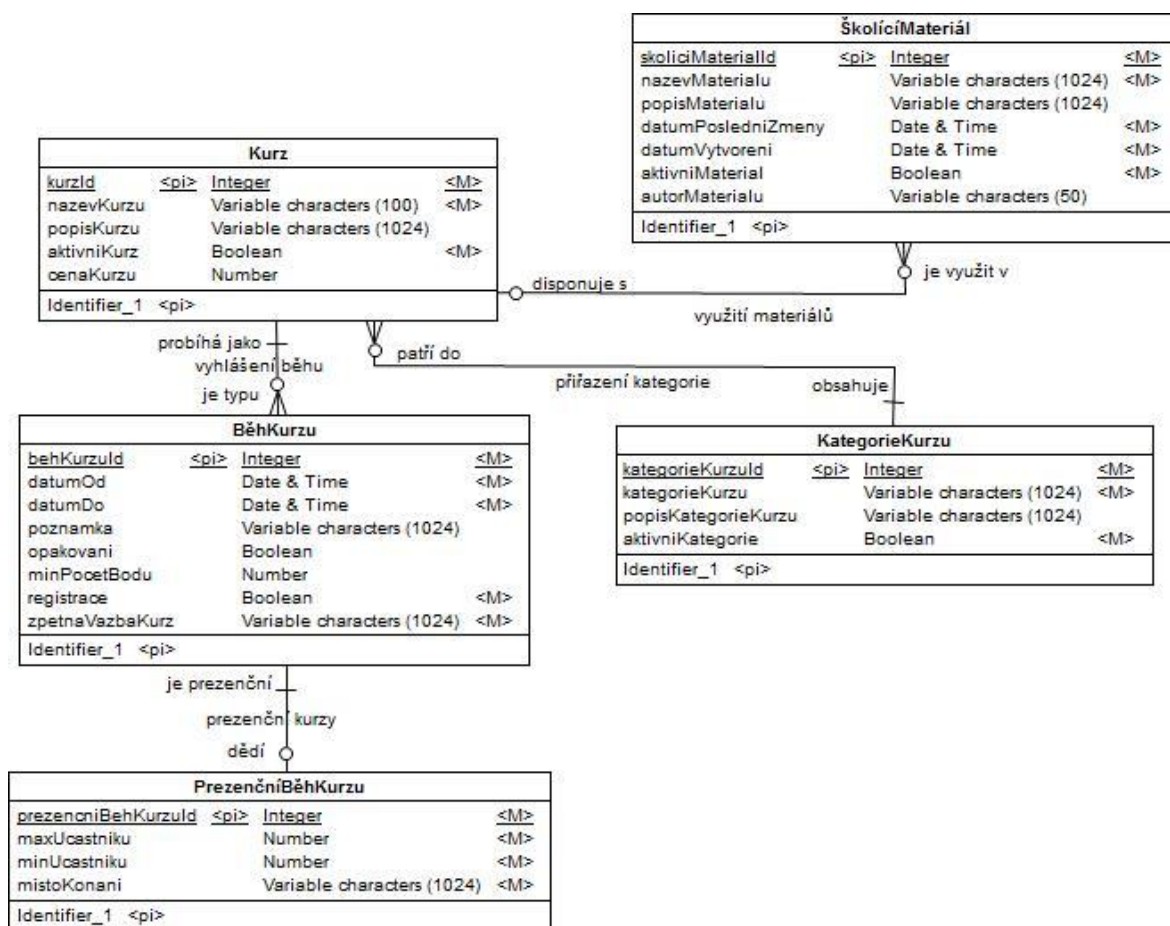
LEKTOR – ŠKOLÍCÍ SPOLEČNOST

Lektor může mít přiřazenou právě jednu školící společnost. Školící společnost může mít přiřazeno více lektorů.

Typ kardinality je 1:M. Nemůže nastat situace, kdy je lektor zaměstnán u více školících společností. Naopak školící společnost určitě bude zaměstnávat více lektorů. Lektor však nemusí být přiřazen k žádné školící společnosti, protože může pracovat jako živnostník.

5.2 Správa kurzů

Tento celek má za účel zajistit evidenci základních údajů o kurzech, které jsou zmíněny v požadavcích. Kurzy nemohou být reprezentovány pouze jednou entitou z důvodu normalizace, ale především z důvodu členitosti kurzů. Kdyby nedošlo ke kategorizaci kurzů a všechny by byly pospolu, nastala by velmi nepřehledná situace s obrovským množstvím dat v jedné tabulce.



Obrázek 7 - Část konceptuálního modelu - Správa kurzů

5.2.1 Entity a jejich atributy

V této části jsou vypsány všechny entity týkající se správy kurzů. Pro každou entitu jsou pak v tabulce znázorněny všechny atributy s formou popisu:

Název atributu – Datový typ – Informace k atributu.

KURZ

Tabulka 12 - Atributy entity Kurz

Název atributu	Datový typ	Informace k atributu
kurzId	Integer	Identifikátor entity
nazevKurzu	VarChar(100)	Název kurzu
popisKurzu	VarChar(1024)	Popis kurzu – specifikuje kurz.
aktivniKurzu	Boolean	Určuje, zda je kurz aktivní
cenaKurzu	Number	Výše ceny kurzu. Jedná se o nepovinný atribut, ne všechny kurzy jsou zpoplatněné. Slouží jako informace pro podnik, ne pro samotné zaměstnance.

KATEGORIEKURZU

Tabulka 13 - Atributy entity KategorieKurzu

Název atributu	Datový typ	Informace k atributu
kategorieKurzuId	Integer	Identifikátor entity
kategorieKurzu	VarChar(1024)	Název kategorie kurzu
popisKategorieKurzu	VarChar(1024)	Popis kategorie kurzu – specifikuje danou kategorii, aby uživatel mohl snáze dohledat požadovaný kurz
aktivniKategorie	Boolean	Určuje, zda je kategorie kurzu aktivní

BĚHKURZU

Tabulka 14 - Atributy entity BehKurzu

Název atributu	Datový typ	Informace k atributu
behKurzuId	Integer	Identifikátor entity
datumOd	Date&Time	Datum zahájení běhu
datumDo	Date&Time	Datum ukončení běhu
poznamka	VarChar(1024)	Poznámka od lektora pro účastníky kurzu
opakovani	Boolean	Určuje, zda je možné kurz opakovat
minPocetBodu	Number	Minimální počet bodů z testu potřebný ke splnění kurzu
registrace	Boolean	Určuje, zda je potřeba se registrovat ke kurzu.
aktivniBehKurzu	Boolean	Určuje, zda je kurz aktivní. Pokud false, pak se nedá přihlašovat k danému běhu.
zpetnaVazbaKurz	VarChar(1024)	Slouží k uchování zpětné vazby uživatele ke konkrétnímu běhu kurzu.

PREZENČNÍBĚHKURZU

Tabulka 15 - Atributy entity PrezenčniBehKurzu

Název atributu	Datový typ	Informace k atributu
prezenčniBehKurzuId	Integer	Identifikátor entity
maxUcastniku	Number	Maximální kapacita běhu
minUcastniku	Number	Minimální kapacita běhu
mistoKonani	VarChar(1024)	Místo, kde probíhá běh

ŠKOLÍCÍMATERIÁL

Tabulka 16 - Atrbuty entity ŠkolícíMateriál

Název atributu	Datový typ	Informace k atributu
skoliciMaterialId	Integer	Identifikátor entity
nazevMaterialu	VarChar(1024)	Název školícího materiálu
popisMaterialu	VarChar(1024)	Popis školícího material
datumPosledniZmeny	Date&Time	Datum, kdy došlo k poslední úpravě material
datumVytvoreni	Date&Time	Datum, kdy došlo k vytvoření materiálu
aktivniMaterial	Boolean	Určuje, zda je material aktivní
autorMaterialu	VarChar(50)	Autor, který materiál přidal

5.2.2 Vztahy mezi entitami

V této části jsou zmíněny vztahy mezi entitami, které zasahují do celku správa kurzů. Pro každý vztah jsou definovány dvě entity, typ vztahu a odůvodnění, proč je zvolen daný typ relace.

KURZ – KATEGORIEKURZU

Kurz musí mít přiřazenou právě jednu kategorii kurzu. Kategorie kurzu může mít přiřazených více kurzů.

Je zvolen typ kardinality 1:M s volitelnou participací na straně kurzu a povinnou u kategorie kurzu. Důvodem je nutnost členit kategorie pro přehlednost a úspěšné vyhledávání kurzu, proto musí každý kurz spadat do určité kategorie.

Samotná kategorie kurzu může obsahovat více kurzů, ale také nemusí. Entita KategorieKuru obsahuje atribut aktivniKategorie datového typu boolean, jehož principem je zneaktivnit všechny kurzy patřící do neaktivní kategorie kurzu. Aplikačně by se poté nastavila viditelnost kategorie/kurzů právě podle hodnoty boolean (vznikne závislost stavu atributu aktivniKurz na stavu atributu aktivniKategorie). Tuto funkčnost nám právě umožnila kardinalita 1:M mezi již zmíněnými entitami.

KURZ – BĚHKURZU

Kurz může mít přiřazen více běhů kurzu. Běh kurzu musí mít přiřazen právě jeden kurz.

Zde je opět typ kardinality 1:M. Lze tak částečně potvrdit, že se opravdu jedná o nejčastěji využívaný typ relace. Tyto dvě entity lze chápat jako abstraktní třídu (Kurz) a dědicí třídu (BěhKurz). Běh kurzu doplňuje specifické informace o kurzu pro jednotlivé běhy, protože každý běh kurzu má odlišné informace. Zatímco entita Kurz v sobě drží informace, které jsou pro běhy konstantní (neměnné). Proto je možné vázat k jednomu kurzu více běhů daného kurzu (lektor může zadat například 3 běhy stejného kurzu po sobě v jeden den).

BĚHKURZU – PREZENČNÍBĚHKURZU

Běh kurzu může mít přiřazen právě jeden prezenční běh kurzu. Prezenční běh kurzu musí mít přiřazen právě jeden běh kurzu.

Typ kardinality je 1:1 s volitelnou participací pro PrezenčníBěhKurz a povinnou participací pro BěhKurz. Entita PrezenčníBěhKurz musí být vázána na BěhKurz, protože obsahuje pouze částečné informace o běhu, které se týkají čistě prezenční účasti. Atributy maxUcastniku, minUcastniku a mistoKonani jsou pro neprezenční běhy zbytečné, tak došlo k jejich separaci. Ovšem zbylé atributy zkopírovat nelze, protože by vznikla redundantní data při každém neprezenčním běhu.

KURZ – ŠKOLÍCÍMATERIÁL

Kurz může mít přiřazen více školících materiálů. ŠkolícíMateriál může být přiřazen na právě jeden Kurz.

Lektoři vytváří specifické školící materiály pro jednotlivé kurzy. Z tohoto důvodu je materiál vázán vždy na jeden typ kurzu. Školící materiál nemusí být přiřazen k žádnému kurzu – důvodem je rozpracovanost materiálu či materiál s neaktuálními daty čekající na úpravu. Veškerou správu materiálů provádí lektor, včetně přiřazení ke kurzu, pro které má také správu.

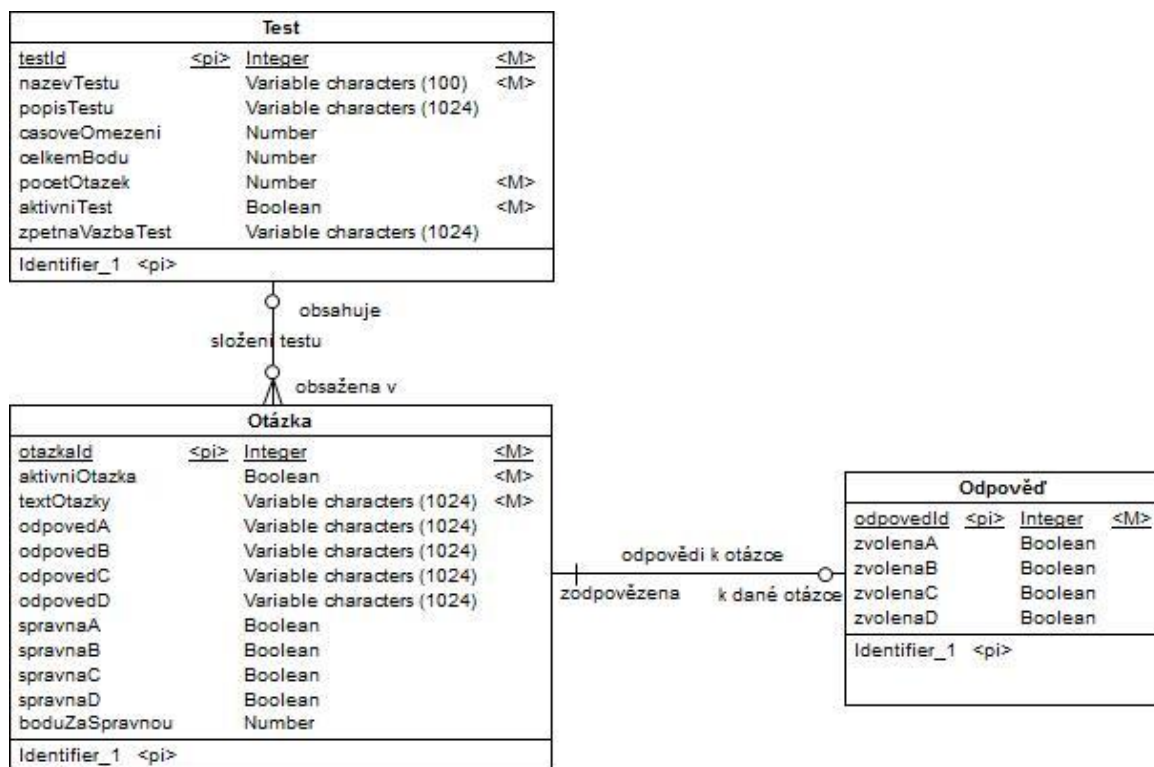
KURZ – ZAMĚSTNANEC

Zaměstnanec se může přihlásit k více kurzům. Na kurz se může zapsat více zaměstnanců.

Typ kardinality je zde M:M, lze se přihlašovat ke kurzům i následně odhlašovat. Pokud se však jedná o prezenční typ kurzu, musí být volná kapacita. Po dosažení maxUchazecu dojde k uzamčení daného běhu.

5.3 Správa testů

Tento celek má za účel zajistit exekuci testů a evidenci jejich výsledků pro dané kurzy. Je stanovena struktura testu vždy v podobě otázky a čtyř možných odpovědí, z nichž jedna nebo více mohou být správné.



Obrázek 8 - Část konceptuálního modelu - Správa testů

5.3.1 Entity a jejich atributy

V této části jsou vypsány všechny entity týkající se správy testů. Pro každou entitu jsou pak v tabulce znázorněny všechny atributy s formou popisu:

Název atributu – Datový typ – Informace k atributu.

TEST

Tabulka 17 - Atributy entity Test

Název atributu	Datový typ	Informace k atributu
testId	Integer	Identifikátor entity
nazevTestu	VarChar(100)	Název testu
popisTestu	VarChar(1024)	Popis testu
casoveOmezeni	Number	Časový limit na dokončení testu

celkemBodu	Number	Celkový počet bodů, který lze v testu dosáhnout
pocetOtazek	Number	Počet otázek obsažených v testu
aktivniTest	Boolean	Určuje, zda je test možné zahájit – závisí především na dostatečném počtu aktivních otázek
zpetnaVazbaTest	Varchar(1024)	Slouží pro zpětnou vazbu uživatele ke konkrétnímu testu.

OTÁZKA

Tabulka 18 - Atributy entity Otázka

Název atributu	Datový typ	Informace k atributu
otazkaId	Integer	Identifikátor entity
otazkaText	Varchar(1024)	Textové zadání otázky
aktivniOtazka	Boolean	Určuje, zda se otázka může vyskytnout v testu
odpovedA	Varchar(1024)	
odpovedB	Varchar(1024)	Textové zadání odpovědi k testové otázce
odpovedC	Varchar(1024)	
odpovedD	Varchar(1024)	
spravnaA	Boolean	Pokud je nastaveno true, pak je daná odpověď správná.
spravnaB	Boolean	
spravnaC	Boolean	Pokud false, odpověď je špatná
spravnaD	Boolean	
boduzaspravnou	Number	Počet bodů za správnou otázku

ODPOVĚĎ

Tabulka 19 - Atributy entity Odpověď

Název atributu	Datový typ	Informace k atributu
odpovedId	Integer	Identifikátor entity
zvolenaA	Boolean	Určuje, jaké odpovědi uživatel zvolil k otázce.
zvolenaB	Boolean	
zvolenaC	Boolean	True=zvolena
zvolenaD	Boolean	False=nezvolena

5.3.2 Vztahy mezi entitami

V této části jsou zmíněny vztahy mezi entitami, které zasahují do celku správa testů. Pro každý vztah jsou definovány dvě entity, typ vztahu a odůvodnění, proč je zvolen daný typ relace.

TEST - UŽIVATEL

Uživatel může zahájit více testů. Test může být přiřazen právě jednomu uživateli.

Typ kardinality je 1:M. Uživatel může spouštět aktivní testy vázané na typ kurzu, ke kterému je uživatel přihlášen. Tato vazba slouží také k evidenci testů vykonaných jednotlivými uživateli. Lze tak zjistit kolik bodů dosáhl z celkového počtu bodů libovolného testu a získat tak statistiku úspěšnosti.

OTÁZKA – ODPOVĚĎ

Na otázku může existovat právě jedna odpověď. Odpověď se váže právě k jedné otázce.

Typ kardinality je 1:1. K otázce může a nemusí existovat odpověď - záleží, zda byla otázka zodpovězena. Zpracovaná odpověď patří pouze k dané otázce, na kterou uživatel odpovídal – nikdy nemůže jednou odpovědí odpovědět na více otázek. Do tabulky Odpověď se zapíše hodnoty odpovědí po dokončení testu, které se pak aplikačně vyhodnotí vůči hodnotám atributů v tabulce Otázka.

TEST – OTÁZKA

K testu může patřit více otázek. Otázka však může být součástí právě jednoho testu.

Každá otázka, kterou lektor připraví, musí být zahrnuta v nějakém testu. Ve výsledku pak existuje množství otázek „n“, vázaných k určitému testu, a při spuštění testu se náhodně vybere počet otázek korespondující s hodnotou atributu pocetOtazek. Pokud nebude k dispozici dostatečný počet otázek pro zahájení testu, aplikačně se ošetří, že atribut aktivniTest bude nabývat hodnoty false.

5.4 Ověření a zobecnění funkčnosti pro další možná uplatnění

Navržené záležitosti jsem ověřil za pomoci programu PowerDesigner, kdy jsem na základě konceptuálního modelu vygeneroval úspěšně fyzický model a základní skripty. Návrh je normalizovaný, integrita je zajištěna a všechny logické vazby jsou vysvětleny v kapitole *Návrh řešení*.

Co se týče zobecnění modelu, lze poměrně snadno rozšířit či zredukovat všechny jeho části. Pokud by jiná firma využívala odlišné uživatelské role, či měla více uživatelských rolí, pak všechny základní uživatelské údaje zůstanou beze změny definovány v entitě Uživatel a analogicky se vytvoří nové entity reprezentující nové role. Vazby by se poté přidaly podle potřeby, resp. podle požadovaných práv dané role.

Dále pokud by se jednalo o malou společnost, která by ke školení chtěla využívat pouze své pracovníky jako interní lektory, lze kvůli přehlednosti z datového modelu odstranit entitu ŠkolícíSpolečnost a všechny její vazby. Není to však potřeba, protože příslušné vazby mají volitelnou participaci a interní lektor může samostatně existovat bez školící společnosti.

Podobně by se nemusela vůbec využít KategorieKurzu, která jak už bylo zmíněno v návrhu řešení, slouží k přehlednosti kurzů. V tomto případě by se však musela plně odstranit i společně s vazbou na Kurz. V současném návrhu je totiž zvolena povinná participace, kdy každý kurz musí spadat právě do určité KategorieKurzu.

Zajímavým návrhem by také zajisté bylo nevytvářet pro každou roli samostatnou entitu, ale řešit vše pomocí parametrů datového typu boolean v entitě Uživatel. Takové řešení rolí by však bylo neflexibilní v momentě, kdy by nastala potřeba zavést nové atributy pro jednotlivé role, například z důvodu uchování konfigurací a parametrů pro zobrazování jednotlivých celků aplikace příslušným uživatelům.

6 Závěr

Kvalifikovaní zaměstnanci jsou nepostradatelným a zároveň nejdražším zdrojem, kterým lze získat převahu nad konkurencí. I přes značnou popularitu současných trendů firemního vzdělávání, se ani zdaleka ne všechny firmy angažují ve zvyšování kvalifikace svých zaměstnanců. Možným důvodem by mohla kromě financí být i obava ze zavedení nového systému.

Hlavním cílem této bakalářské práce bylo navrhnout přijatelné řešení databázového modelu v kontextu zvyšování kvalifikace zaměstnanců. Možná ještě důležitější než samotný návrh byla v tomto případě analýza současného stavu a následná definice požadavků a potřeb týkající se této problematiky.

První kapitola shrnuje nejdůležitější teoretické principy z oblasti relačně datového modelování, nutné k pochopení metodologie návrhu databáze. Zvýšená pozornost je věnována zejména datové normalizaci, která je základním kamenem datového modelování.

Stěžejní informace pro návrh konceptuálního datového modelu jsou obsaženy v části druhé, obsahující specifické relevantní požadavky na chování systému. Jako nedostatky současného vzdělávání vidím především chybějící zpětnou vazbu mezi zaměstnanci a lektory při online kurzech. Pro zefektivnění vzdělávacího procesu jsem tak zavedl povinnou zpětnou vazbu pro všechny absolvované kurzy a nepovinnou zpětnou vazbu pro jednotlivé testy. Další nedostatek pak shledávám v nízké integraci osobních dat účastníků s absolvovanými kurzy a jejich vyhodnocením. Požadovaným zlepšením by určitě mohly být rozsáhlejší reporty o zaměstnancích a jejich osobním rozvoji.

Ve třetí části se pak nachází již zmíněné hlavní řešení korespondující s identifikovanými požadavky. Výsledný konceptuální datový model je ve formě přílohy na konci práce.

7 Seznam použitých zdrojů

- 1 CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7.
- 2 HERNANDEZ, Michael J. *Návrh databází*. 1. vyd. Praha: Grada, 2006, 408 s. ISBN 80-247-0900-7.
- 3 JAROSLAV, Pokorný. *Databázové systémy*. Vyd. 2. Praha: ČVUT, 2004, 148 s. ISBN 80-01-02789-9.
- 4 OPPEL, Andrew. *Databáze bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2006, 319 s. ISBN 80-251-1199-7.
- 5 Lucidchart [online], Dostupný z: <http://lucidchart.com/>.
- 6 ARMSTRONG, A. *Řízení lidských zdrojů*. Praha: Grada, 2002.
- 7 KOUBEK, Josef. *Řízení lidských zdrojů: základy moderní personalistiky*. 5., rozš. a dopl. vyd. Praha: Management Press, 2015, 399 s. ISBN 978-80-7261-288-8.
- 8 Firmy na školeních nešetří, ale vydaje více promyšlejí [online]. 2014. Dostupné z: <http://byznys.ihned.cz/c1-63155770-firmy-na-skolenich-nesetri-ale-vydaje-vice-promysleji>
- 9 Important eLearning Statistics for 2013 [online]. 2014. Dostupné z: certifyme.net/osha-blog/elearning-statistics

8 Seznam obrázků

Obrázek 1 - Use case diagram - ukázka.....	17
Obrázek 2 - Znázornění uživatelů v Use case diagramech.....	24
Obrázek 3 - Use case diagram pro uživatelský pohled Správce.....	28
Obrázek 4 - Use case diagram pro uživatelský pohled Zaměstnance.....	29
Obrázek 5 - Use case diagram pro uživatelský pohled Lektor.....	30
Obrázek 6 - Část konceptuálního modelu - Správa osob.....	33
Obrázek 7 - Část konceptuálního modelu - Správa kurzů.....	37
Obrázek 8 - Část konceptuálního modelu - Správa testů.....	41

9 Seznam tabulek

Tabulka 1 - Notace vraních stop pro ER modelování ^[1]	19
Tabulka 2 – Role uživatelů a jejich popis	24
Tabulka 3 - Popis use case diagramu pro uživatelský pohled Správce Chyba! Záložka není definována.	
Tabulka 4 - Popis use case diagramu pro uživatelský pohled Zaměstnanec Chyba! Záložka není definována.	
Tabulka 5 - Popis use case diagramu pro uživatelský pohled Lektor.. Chyba! Záložka není definována.	
Tabulka 6 - Atributy entity Uživatel	Chyba! Záložka není definována.
Tabulka 7 - Atributy entity Zaměstnanec	Chyba! Záložka není definována.
Tabulka 8 - Atributy entity Správce	Chyba! Záložka není definována.
Tabulka 9 - Atributy entity Lektor	Chyba! Záložka není definována.
Tabulka 10 - Atributy entity ŠkolícíSpolečnost	Chyba! Záložka není definována.
Tabulka 11 - Atributy entity Kontakt	Chyba! Záložka není definována.
Tabulka 12 - Atributy entity Kurz	Chyba! Záložka není definována.
Tabulka 13 - Atributy entity KategorieKurzu	Chyba! Záložka není definována.
Tabulka 14 - Atributy entity BehKurzu	Chyba! Záložka není definována.
Tabulka 15 - Atributy entity PrezenčníBehKurzu	Chyba! Záložka není definována.
Tabulka 16 - Atributy entity ŠkolícíMateriál	Chyba! Záložka není definována.
Tabulka 17 - Atributy entity Test	Chyba! Záložka není definována.
Tabulka 18 - Atributy entity Otázka	Chyba! Záložka není definována.
Tabulka 19 - Atributy entity Odpověď	Chyba! Záložka není definována.

PŘÍLOHA A – KONCEPTUÁLNÍ DATOVÝ MODEL

