VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY
INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE PRO DROBNÉ CHOVATELE ZVÍŘAT

DATABASE DESIGN FOR SMALL ANIMAL BREEDERS

DIPLOMOVÁ PRÁCE
MASTER´S THESIS

AUTOR PRÁCE Bc. Zdeněk Záboj
AUTHOR

VEDOUCÍ PRÁCE Ing. Jiří Kříž, Ph.D.
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	**Bc. Zdeněk Záboj**
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	**Ing. Jiří Kříž, Ph.D.**
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh databáze pro drobné chovatele zvířat

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je identifikovat klíčové oblasti, ve kterých drobní chovatelé mohou využívat databázový systém pro evidenci chovaných zvířat a s nimi souvisejících položek. Na základě zjištěných informací následně navrhnout databázový systém, který bude tyto potřeby reflektovat.

Základní literární prameny:

CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KROENKE, David, David J. AUER a Jakub GONER. Databáze. Brno: Computer Press, 2015. 496 s. ISBN 978-80-251-4352-0.

LAURENČÍK, Marek. SQL: podrobný průvodce uživatele. Praha: Grada Publishing, 2018. 216 s. ISBN 978-80-271-0774-2.

LEISS, Oliver a Jasmin SCHMIDT. PHP v praxi: pro začátečníky a mírně pokročilé. Praha: Grada, 2010. 242 s. ISBN 978-80-247-3060-8.

OPPEL, Andrew J. SQL bez předchozích znalostí: [průvodce pro samouky]. Brno: Computer Press, 2008. 240 s. ISBN 978-80-251-1707-1.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. RNDr. Bedřich Půža, CSc. ředitel	doc. Ing. et Ing. Stanislav Škapa, Ph.D. děkan

Abstrakt

Tato diplomová práce je zaměřena na návrh databáze pro drobné chovatele, pro které nemá smysl využívat rozsáhlá podniková řešení. V této práci se zaměřím na analýzu potřeb těchto subjektů a na návrh databáze, která tyto potřeby adekvátně pokryje s minimálními finančními náklady.

Abstract

This master´s thesis is focused on the design of a database for small breeders, for whom it makes no sense to use large enterprise solutions. In this thesis I will focus on the analysis of the needs of those individuals and on the design of a database that will satisfy their needs with minimal financial requirements.

Klíčová slova

databázový systém, databáze, MySQL, XAMPP, drobní chovatelé

Key words

database system, database, MySQL, XAMPP, small breeders

Bibliografická citace

ZÁBOJ, Zdeněk. *Návrh databáze pro drobné chovatele zvířat* [online]. Brno, 2020 [cit. 2020-05-15]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/127644. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jiří Kříž.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 15. května 2020

……

podpis autora

Poděkování

V první řadě bych chtěl poděkovat vedoucímu této práce Ing. Jiřímu Křížovi, Ph.D. za konzultace a cenné rady udělené během zpracovávání této diplomové práce, a to zejména v době ztížených podmínek studia. Dále bych chtěl poděkovat rodině a blízkým za podporu a motivaci nejen během tvorby této práce ale i v průběhu celého studia.

OBSAH

ÚVOD 11

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ 13

1 TEORETICKÁ VÝCHODISKA PRÁCE 15

1.1 Obecné pojmy 15

1.1.1 Informace 15

1.1.2 Data 16

1.1.3 Znalosti 17

1.2 Databázový systém 17

1.2.1 Databázové aplikace 18

1.2.2 Systém řízení databáze (SŘBD) 18

1.2.3 Databáze 19

1.3 Životní cyklus vývoje databázových systémů 19

1.4 Základní pojmy datového modelování 22

1.5 Typy datových modelů 22

1.5.1 Lineární datový model 23

1.5.2 Hierarchický datový model 23

1.5.3 Síťový datový model 24

1.5.4 Relační datový model 24

1.5.5 Objektový datový model 30

1.6 Metodologie návrhu databáze 31

1.6.1 Konceptuální návrh databáze 31

1.6.2 Logický návrh databáze 32

1.6.3 Fyzický návrh databáze 33

1.7 Proces normalizace 34

1.7.1 První normální forma 34

1.7.2 Druhá normální forma 34

1.7.3 Třetí normální forma ... 35
1.7.4 Boyce-Coddova normální forma ... 35
1.7.5 Čtvrtá normální forma ... 35
1.7.6 Pátá normální forma ... 36
1.8 Jazyk SQL ... 36
1.9 Datové typy ... 38
1.10 ER diagram ... 39
1.11 XAMPP ... 39
2 ANALÝZA SOUČASNÉHO STAVU ... 40
2.1 Stávající způsob evidence ... 40
2.2 Chov ovcí a koz ... 40
2.3 Chov psů ... 42
2.4 Chov drůbeže ... 44
2.5 Evidence nákladů a výnosů spojených s chovem ... 45
2.6 Krmení ... 46
2.7 Evidence zásob ... 46
2.8 Shrnutí požadavků na databázi ... 46
2.9 Dostupné platformy ... 47
2.9.1 Oracle ... 47
2.9.2 MySQL ... 47
2.9.3 Microsoft SQL Server ... 48
2.9.4 PostgreSQL ... 48
2.9.5 MongoDB ... 48
2.9.6 MariaDB ... 49
2.9.7 DB2 ... 49
2.9.8 SAP HANA ... 49
2.10 Výběr platformy ... 50
3 VLASTNÍ NÁVRHY ŘEŠENÍ ... 51

3.1 Základní vymezení navrhovaného řešení ... 51
3.2 Konceptuální návrh databáze ... 52
3.2.1 Identifikace základních entit ... 52
3.2.2 Identifikace základních vztahů mezi entitami ... 53
3.2.3 Grafické znázornění základních vztahů ... 55
3.2.4 Identifikace atributů základních entit ... 56
3.3 Logický návrh databáze ... 60
3.3.1 Návrh jednotlivých tabulek ... 60
3.4 ER diagram ... 96
3.5 Fyzický návrh databáze ... 97
3.5.1 Vytvoření jednotlivých tabulek ... 97
3.5.2 Ukázka vytvoření tabulky ... 99
3.5.3 Návrh vybraných pohledů, triggeru a procedury ... 100
3.5.4 Zálohy databáze ... 108
3.6 Další kroky před dodáním finálního řešení ... 109
3.7 Přínosy navrhovaného řešení ... 109
ZÁVĚR ... 110
SEZNAM POUŽITÉ LITERATURY ... 111
SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ ... 113
SEZNAM OBRÁZKŮ ... 114
SEZNAM TABULEK ... 115
SEZNAM PŘÍLOH ... 117

ÚVOD

V současné době je dostupné velké množství odlišných databázových systémů určených k různému typu užití, které se liší například předpokládaným počtem uživatelů. Tyto běžně dostupné systémy pokrývají rozličné funkcionality a jsou k dispozici v různých cenových relacích.

V rámci svého okolí jsem identifikoval drobné chovatele domácích a hospodářských zvířat, kteří v současné době používají především papírovou evidenci, která jim umožňuje provést pouze základní evidenci agendy spojené s chovem, a digitalizace tohoto obsahu je prováděna výhradně s využitím kancelářského softwaru (Word, či Excel).

Takováto evidence je poměrně komplikovaná, náchylná na chybovost a nesoulad ve strukturách jednotlivých evidovaných dat, kromě jiného ztěžuje, či zcela znemožňuje rozsáhlejší vyhodnocování důležitých ukazatelů, neboť data nejsou vhodně strukturovaná, některé informace jsou uloženy pouze v myslích chovatelů a mohou být po čase zapomenuty, či značně zkresleny.

Tito drobní chovatelé ve většině případů chovají menší množství zvířat (spíše desítky) a v některých ohledech nevyžadují tak podrobnou evidenci jako rozsáhlé zemědělské podniky. Zpravidla se jedná o osoby, které nejsou technicky zdatné a analytické nástroje dostupné v rámci zmiňovaných softwarů jsou pro ně nekomfortní.

Pro subjekty, kterým má sloužit navrhované řešení, není chovatelská činnost primárním zdrojem příjmu, ale spíše volnočasovou aktivitou, a i z tohoto důvodu nechtějí do databázového řešení investovat velké množství prostředků.

Z výše uvedených důvodů spatřuji velký potenciál v možnosti navrhnout pro takovéto subjekty databázové řešení, se kterým budou moci pracovat pomocí webového rozhraní. Návrh databázového řešení, které odpovídá požadavkům drobných chovatelů, je předmětem této práce.

Návrh databáze, který je uveden v této práci, je pouze prvním krokem vytvoření komplexního řešení pokrývajícího evidenční potřeby drobných chovatelů. Celé řešení předpokládá dodatečné vytvoření uživatelského rozhraní dostupného prostřednictvím webu, které bude pro tyto subjekty optimálně navržené a bude snadno přístupné i při využití mobilních zařízení.

Tato práce se nicméně zaměřuje pouze na návrh potřebných datových struktur a následné vytvoření zmiňovaného webového rozhraní, či jeho optimalizace přesahují rámec této práce. Nicméně webové uživatelské rozhraní je základním kamenem celého konceptu komplexního řešení a je zdrojem jeho hlavních přínosů, proto je vhodné se o něm zmínit, byť v rámci této práce nebude řešeno.

Potvrzením potenciálu takovéhoto řešení je i fakt, že od doby, kdy jsem se návrhem začal zabývat a provedl analýzu s prvními drobnými chovateli ve svém okolí, již o řešení projevilo zájem několik dalších subjektů.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Cílem práce je identifikovat klíčové oblasti, ve kterých drobní chovatelé mohou využívat databázový systém pro evidenci chovaných zvířat a s nimi souvisejících položek. Na základě zjištěných informací následně bude navržen databázový systém, který bude tyto potřeby reflektovat.

Tato práce bude členěna do několika hlavních kapitol. Nejprve budou představena teoretická východiska pro tuto práci, následně bude provedena analýza potřeb drobných chovatelů a dostupných platforem a na základě těchto informací bude následně navrženo vhodné řešení.

Teoretická část této práce bude zaměřena na objasnění obecných pojmů z oblasti návrhu databází, jako jsou data, či informace, popis jednotlivých složek databázového systému a jednotlivé datové modely. Největší důraz je kladen na teorii spojenou s relačními databázemi, jelikož navrhované řešení bude z těchto poznatků čerpat.

V analytické části této práce bude provedena analýza potřeb drobných chovatelů, která bude zaměřena zejména na specifikaci, jaká zvířata jsou těmito chovateli běžné chovaná, na identifikaci dat, které tito chovatelé mohou chtít evidovat, a na definování analytických výstupů, které by chovatelé v systému rádi viděli a které by jim umožnili efektivně plánovat chov.

Pro zjištění potřeb drobných chovatelů bude použita metoda pozorování, která poskytne nejvěrohodnější informace o běžném provozu těchto subjektů, a metoda rozhovorů, která pozorované skutečnosti doplní o vysvětlení souvislostí.

Vzhledem k tomu, že v současné době je na trhu k dispozici velké množství platforem, které lze pro tvorbu řešení využít, bude v rámci této části práce rovněž provedeno jejich srovnání a bude vybrána platforma, na které bude řešení realizováno.

Návrhová část této práce bude členěna do tří bloků, které odpovídají metodologii návrhu databáze. Bude se jednat o konceptuální, logický a fyzický návrh.

V rámci konceptuálního návrhu bude provedena identifikace základních entit, které vyplynuly z analýzy potřeb drobných chovatelů, a jejich vzájemných vztahů, které bude nutné v rámci databáze reflektovat. Pro identifikované entity budou následně stanoveny jednotlivé atributy, které vyplývají z jejich vlastností v reálném světě.

V rámci logického návrhu budou entity identifikované v předchozí fázi návrhu převedeny do jednotlivých databázových tabulek a budou doplněny o detailní přehled polí, které budou obsahovat. U jednotlivých polí bude určeno, jakých hodnot mohou nabývat, a bude stanovena povinnost zadání jejich hodnot. Dále budou určeny primární a cizí klíče tabulek. Na závěr bude vytvořen ER diagram představující všechny vzájemné vztahy tabulek zahrnutých v navrhovaném řešení a jejich polí.

V poslední fázi návrhu databáze, tedy v rámci fyzického návrhu, budou vytvořeny tabulky definované v předchozích fázích ve vybraném prostředí, bude provedeno jejich naplnění testovacími daty a budou představeny vybrané možnosti využití takto vytvořené databáze. V neposlední řadě budou v krátkosti popsány možnosti zálohování navrhované databáze. a to jak v průběhu vývoje řešení, tak následně po nasazení do ostrého provozu.

Na závěr bude provedeno zhodnocení přínosů takto navrženého řešení pro drobné chovatele.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Následující podkapitoly obsahují teoretická východiska této práce. První podkapitola se zaměřuje na definici obecných pojmů, mezi které lze bezesporu zařadit data, informace a znalosti. Další podkapitoly obsahují popis jednotlivých komponent databázového systému, jednotlivých datových modelů, zejména pak relačního datového modelu, životního cyklu databáze, metodologie návrhu databáze, procesu normalizace a jednotlivých normálních forem. Kromě jiného bude představen ER diagram sloužící pro grafickou reprezentaci vztahů mezi entitami a bude představeno prostředí XAMPP, které bude použito při fyzickém návrhu databáze a s ním související jazyk SQL, či datové typy.

1.1 Obecné pojmy

Jak již bylo zmíněno výše, základní obecné pojmy spojené s problematikou databází jsou informace, data a znalosti.

1.1.1 Informace

Dle Druckera produkují klasické výrobní faktory, kterými jsou práce, půda a kapitál, klesající výnosy a jediným smysluplným výrobním faktorem jsou informace, znalosti a vědomosti. V postkapitalistické společnosti jsou třídou nahrazující kapitalisty a proletáře kvalifikovaní pracovníci, kteří pro svou práci využívají znalosti, informace a vědomosti (1, s. 13, 163).

Pojem informace lze definovat různými způsoby. V pojetí některých autorů je informace součástí zpracovatelského řetězce, který se skládá z následující posloupnosti:

- reálný svět,
- data,
- informace
- a znalosti (2, s. 22).

„Informace je pojmenování pro obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním.“ (3, s. 13)

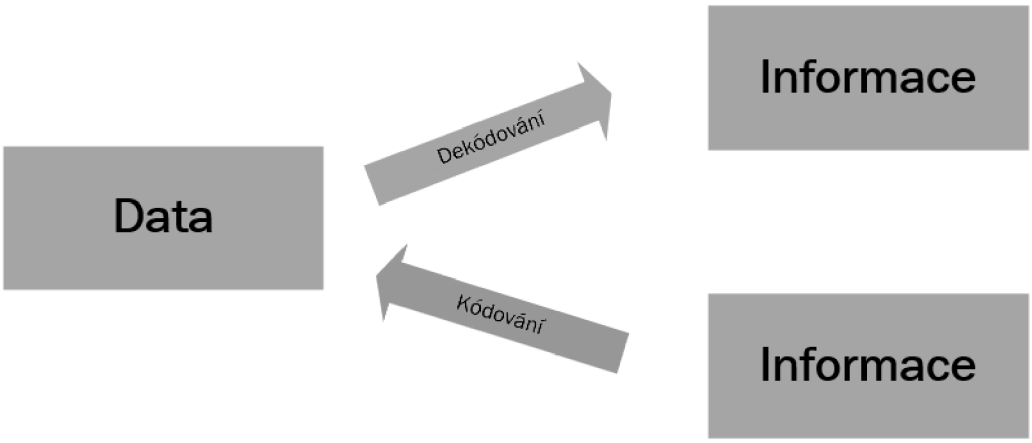
Informaci lze alternativně definovat jako data, která prošla zpracováním, či získala strukturu, díky které dostává význam pro jednotlivce nebo organizaci (4, s. 36).

Na informaci jako takovou, lze nahlížet z několika pohledů — syntaktického, sémantického a pragmatického. První dva pohledy nahlíží na informaci bez ohledu na samotného příjemce této informace. Syntaktický pohled nahlíží na vnitřní stavbu informace, zatímco sémantický pohled se zaměřuje na obsah konkrétní informace. Pragmatický pohled pracuje s významem informace pro konkrétního příjemce (5, s. 19).

1.1.2 Data

Data jsou nezpracovaná fakta mající pro jednotlivce, či organizaci určitou důležitost (4, s. 36).

Data jako taková, bez jakéhokoliv kontextu, popisují vlastnosti objektů reálného světa a je jim běžně přisuzován význam zpráv. Pokud jsou data začleněná do kontextu tak, že mají význam a smysl, stávají se informacemi (3, s. 48; 6, s. 5).



Obr. 1:Vztah mezi informacemi a daty

(Zdroj: 6, s. 5)

1.1.3 Znalosti

Znalost jako takovou lze popsat jako informaci, která je chápána v kontextu s dalšími informacemi. Pokud na znalosti aplikujeme například intuici, zkušenosti, morální hodnoty a principy (například to, co považujeme za dobré a špatné konání), můžeme se dostat do stavu, který lze označit za moudrost (2, s. 63, 64).

1.2 Databázový systém

„Kolekce databázových aplikací, které interagují s databází spolu, DMBS a databáze samotná.“ (4, s. 39)

Hlavní komponenty databázového systému jsou:

- uživatel,
- databázová aplikace,
- systém řízení databáze
- a databáze (7, s. 29).

Uživatelé mohou prostřednictvím databázové aplikace a systému řízení databáze pracovat se záznamy uloženými v databázi. Databázová aplikace je jeden nebo více počítačových programů, který uživatelům zobrazuje data ve formě formulářů a sestav a komunikuje se systémem řízení databáze prostřednictvím SQL příkazů. Systém řízení databáze je počítačový program, jehož prostřednictvím lze jednotlivé databáze vytvářet a pracovat s nimi. Vztahy jednotlivých komponent představuje následující obrázek (7, s. 29).

Obr. 2: Vztahy komponent databázového systému

(Zdroj: vlastní zpracování dle 7, s. 30)

1.2.1 Databázové aplikace

„Počítačový program interagující s databází vyvoláním odpovídajícího požadavku (typicky jednoho nebo více příkazů SQL) pro DBMS.“ (4, s. 39)

Jak již bylo zmíněno výše, databázové aplikace stojí mezi uživatelem a systémem řízení databáze. Tyto programy si organizace vytvářejí samy, nebo využívají řešení dodávané softwarovými firmami. Mezi hlavní funkce těchto aplikací patří:

- tvorba a zpracování formulářů a sestav,
- zpracování jednotlivých uživatelských dotazů,
- provádění aplikační logiky a řízení aplikace (7, s. 29, 32).

1.2.2 Systém řízení databáze (SŘBD)

„Softwarový systém, který uživateli umožňuje definovat, vytvářet a udržovat databázi a poskytuje řízený přístup k této databázi.“ (4, s. 38)

Mezi hlavní funkce systému řízení databáze patří:

- tvorba samotné databáze, tabulek a podpůrných struktur a jejich údržba,
- vkládání, upravování a mazání dat v databázi,
- čtení dat uložených v databázi,
- zálohování a obnova databáze,
- vynucování jednotlivých pravidel,
- zajištění zabezpečení (7, s. 31).

1.2.2.1 Architektury SŘBD

Před vytvořením a rozšířením webu byla využívána dvouvrstvá architektura klient – server. Strana klienta obstarávala uživatelské rozhraní a řešila hlavní logiku provozu, na straně serveru byl poté řízen přístup k databázi. Toto řešení začalo být vzhledem k velkým nárokům na zdroje na straně klienta postupně nahrazováno třívrstvou architekturou. Tato architektura je složena z tří vrstev:

- vrstvy uživatelského rozhraní (tenký klient zajišťující uživatelské rozhraní),
- vrstvy logiky provozu (střední vrstva označovaná jako aplikační server),
- DBMS (databázový server uchovávající data pro aplikační server) (4, s. 41-43).

Mezi výhody této architektury patří snížení nároků na hardware na straně klienta, zjednodušení údržby aplikací, snadnější změny jednotlivých vrstev a optimalizace zatížení (4, s. 43).

1.2.3 Databáze

Databázi lze chápat jako jediné, či velké uložiště dat, ke kterému může mít v jednom okamžiku přístup velký počet uživatelů. Databáze je většinou zdrojem sdíleným napříč celou organizací a jejím vlastníkem není konkrétní uživatel, či oddělení organizace (4, s. 37).

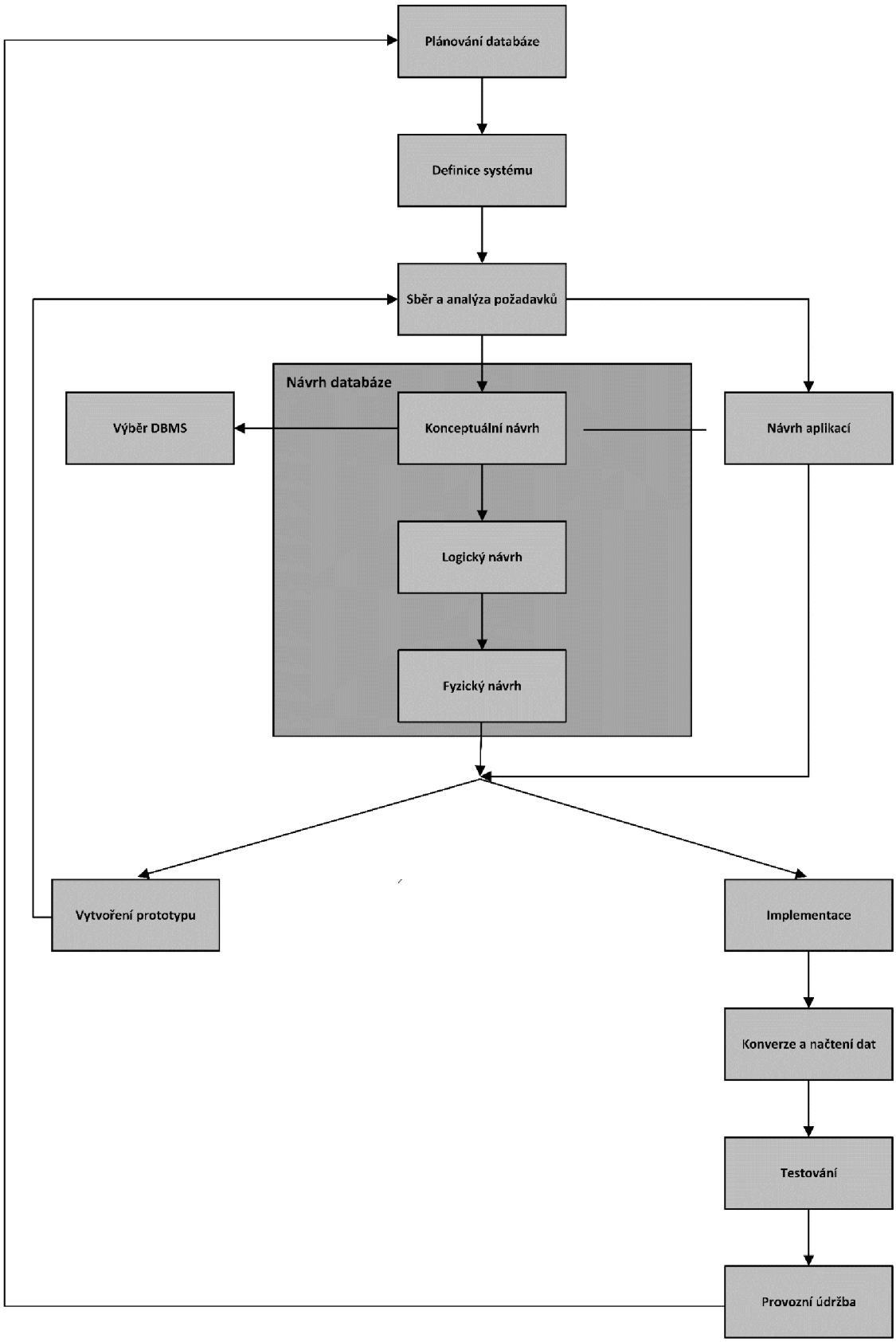
V případě relačních databází lze databázi definovat jako vzájemně propojené tabulky mající vlastní popis, které obsahují jednotlivé záznamy. Díky tomu, že databáze obsahuje popis své vlastní struktury, je možné s ní pracovat bez nutnosti zjišťovat potřebné informace mimo tuto databázi (7, s. 30).

Data popisující strukturu databáze se nazývají metadata. Mezi metadata patří například názvy jednotlivých tabulek, názvy sloupců, které tyto tabulky obsahují, vlastnosti tabulek a jiné (7, s. 31).

Kromě samotných dat a metadat databáze obsahují indexy, či aplikační metadata (7, s. 31).

1.3 Životní cyklus vývoje databázových systémů

Životní cyklus vývoje databázového systému se skládá z několika fází a jeho složitost závisí na velikosti navrhovaného systému. Jednotlivé kroky po sobě nemusí bezpodmínečně následovat tak, jak je uvedeno na následujícím obrázku, některé části mohou být realizovány současně a může též docházet k opakování některých kroků na základě zpětné vazby (4, s. 109).



Obr. 3: Životní cyklus vývoje databázového systému

(Zdroj: 4, s. 110)

V rámci prvního kroku, tedy v rámci plánování databáze, jsou stanoveny hlavní a dílčí cíle databázového systému. Dále je vhodné odhadnout rozsah práce potřebné pro realizaci databáze a stanovit potřebné zdroje a finanční prostředky (4, s. 109).

Následujícím krokem je definice databázového systému, během níž je nutné určit rozsah a hranice navrhovaného řešení a identifikovat jednotlivé uživatelské pohledy, neboť každá skupina uživatelů může mít na systém jiné požadavky (4, s. 111-112).

Dalším krokem je sběr a analýza požadavků na databázový systém. Tento krok je nesmírně důležitý pro samotný návrh databáze, neboť špatně určené požadavky mohou vést k vypracování návrhu, který nebude splňovat očekávání uživatelů, což může vést k odmítnutí navrhovaného řešení (4, s. 112-113).

Tématem návrhu databáze se budu blíže zabývat v kapitole 1.6 Metodologie návrhu databáze. Samotný návrh databáze je tvořen konceptuálním, logickým a fyzickým návrhem databáze. Během první fáze jsou identifikovány důležité entity a jejich vzájemné vztahy. Následně v druhé fázi jsou navrženy jednotlivé tabulky, které představují strukturu, ve které budou data popisující tyto entity ukládána. Ve třetí fázi jsou tyto tabulky fyzicky vytvořeny (4, s. 115).

Kromě samotného návrhu databáze, je v případě, že organizace nevyužívá žádný DBMS, vhodné jeho výběr provést po dokončení logického návrhu, nicméně tento výběr může proběhnout i dříve, a to v případě, že jsou známy potřebné systémové požadavky organizace (4, s. 115-116).

Návrh aplikací probíhá současně s návrhem databáze. V rámci tohoto kroku je nutné zajistit splnění všech požadavků identifikovaných během fáze sběru a analýzy. Je nutné navrhnout způsob interakce uživatelů s uloženými daty a uživatelské rozhraní pro práci s databází (4, s. 116-117).

Na základě návrhu databáze může být vytvořen prototyp, který neobsahuje všechny požadované funkcionality. Tento prototyp může být využit k představení navrhovaného řešení uživatelům. Na základě jejich zpětné vazby mohou být některé funkcionality upraveny do podoby, která bude těmto uživatelům lépe vyhovovat, případně mohou být nalezeny nové funkcionality, které jsou potřebné, ale nebyly zahrnuty v původním návrhu (4, s. 118).

Po ukončení kroků souvisejících s návrhem následuje fáze implementace navrhovaného řešení. Během této fáze jsou implementovány databázové struktury, uživatelské pohledy, transakce, databázové aplikace a jiné (4, s. 118-119).

V případě, že nově navrhovaný systém nahrazuje stávající řešení, je nutné provést konverzi dat do požadovaného formátu a provést jejich načtení (4, s. 119).

Dalším krokem je testování navržené databáze a databázových aplikací. Cílem tohoto procesu je nalézt chyby, které by v budoucnu vedly k problémům při ostrém provozu (4, s. 119).

Posledním krokem životního cyklu vývoje databázového systému je provozní údržba, v jejímž rámci je prováděno monitorování výkonu, udržování a případně je provedena aktualizace databázového systému (4, s. 120).

1.4 Základní pojmy datového modelování

Reprezentace každého objektu reálného světa je v rámci datového modelu tvořena datovým objektem. Pro každý takovýto objekt lze definovat jednotlivé atributy, které tento objekt popisují. Struktura jednotlivých atributů konkrétního objektu je nazývána větou (8, s. 15).

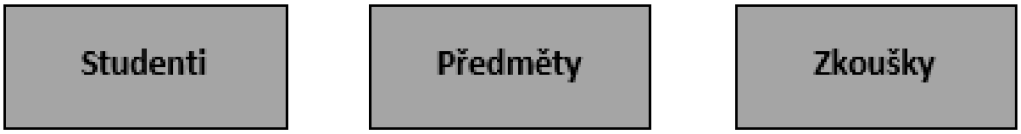
1.5 Typy datových modelů

Existuje několik typů datových modelů, které lze při tvorbě databáze využít. Vzhledem k tomu, že datové objekty jsou obrazem objektů reálného světa, existují mezi nimi stejně jako v reálném světě vazby. Struktura vět jednotlivých datových objektů je pro každý objekt jiná a je třeba ji datovým modelem zachytit. Datové modely lze rozdělit na:

- lineární,
- hierarchický,
- síťový,
- relační,
- objektový (8, s. 21).

1.5.1 Lineární datový model

Lineární datový model je jediný model, pro který lze využít libovolné médium. V rámci tohoto modelu neexistují žádné vztahy mezi objekty a mezi jednotlivými větami, tudíž existuje pouze vztah předchůdce a následovník. Příkladem realizace tohoto modelu může být například kartotéka (8, s. 21-22).

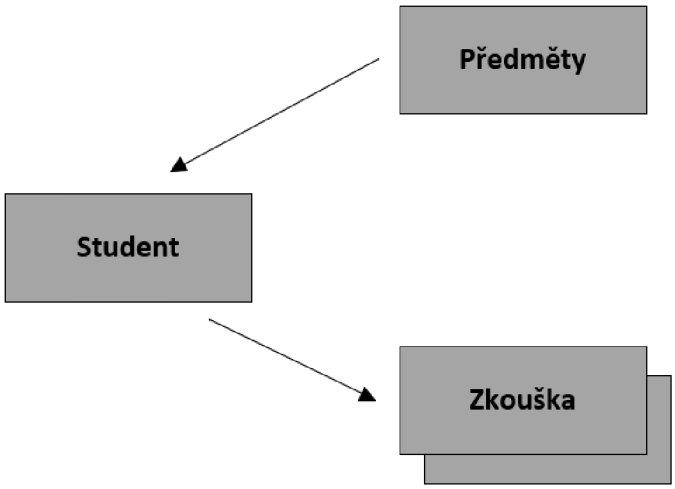


Obr. 4: Ukázka lineárního datového modelu

(Zdroj: vlastní zpracování dle 8, s. 21)

1.5.2 Hierarchický datový model

Hierarchický datový model tvoří rodičovské věty, které jsou pomocí pointerů provázány na jim podřízené věty, které mají jinou strukturu a obsah. Tento vztah se nazývá rodič/potomek. Na obsah podřízených vět se lze dostat pouze prostřednictví rodičovských segmentů, kterým tyto věty náleží (8, s. 22).



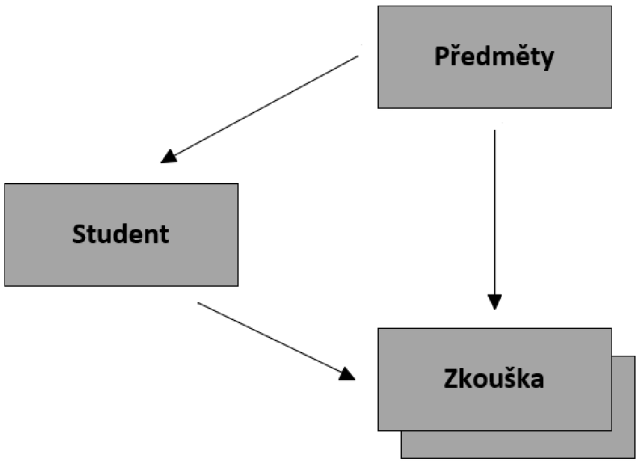
Obr. 5: Ukázka hierarchického datového modelu

(Zdroj: vlastní zpracování dle 8, s. 22)

Výhodou tohoto modelu je rychlost vyhledávání, nicméně za významnou nevýhodu lze považovat dobu nutnou pro reorganizaci souborů při jejich vkládání, či k rekonstrukci dat, pokud dojde k poškození databáze (8, s. 22).

1.5.3 Síťový datový model

Síťový model lze považovat za rozšíření hierarchického modelu. V tomto modelu již vazbu neoznačujeme vazbou rodič/potomek. Jednotlivé pointery mohou vést mezi libovolnými segmenty, což je hlavní výhodou tohoto modelu (8, s. 23).



Obr. 6: Ukázka síťového datového modelu

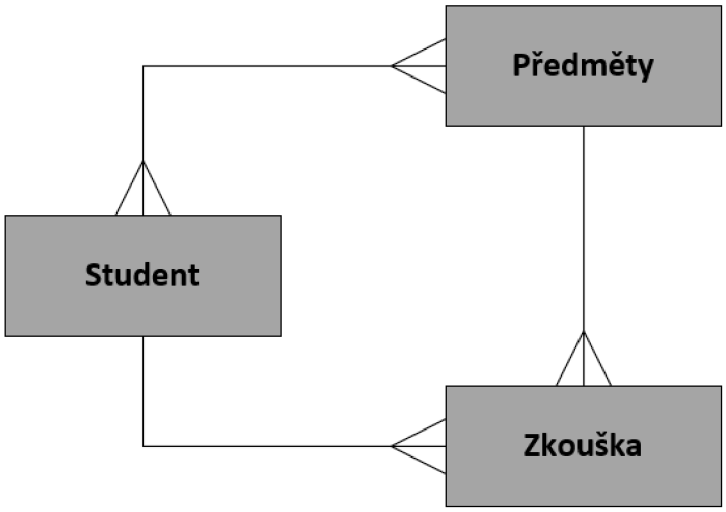
(Zdroj: vlastní zpracování dle 8, s. 23)

1.5.4 Relační datový model

Relační datový model byl v roce 1970 navržen Edgarem F. Coddem, který jej představil ve své práci s názvem Relační model dat pro velké sdílené databanky. Tato práce je v dnešní době považována za průlomovou v oblasti databázových systémů (4, s. 62).

V současné době se jedná o nejpoužívanější datový model. Tento model je spojen z více lineárních modelů, mezi kterými je propojení řešeno pomocí relačních klíčů. Toto propojení vzniká v případě, kdy vznikne potřeba práce s daty z různých tabulek, a zaniká v momentě ukončení této práce (8, s. 24).

Tento model byl vytvořen s ohledem na požadované cíle, které měl splňovat, kterými byl zejména požadavek na vysoký stupeň nezávislosti dat a zvládnutí problémů spojených s konzistencí, redundancí a sémantikou dat (4, s. 62).



Obr. 7: Ukázka relačního datového modelu

(Zdroj: vlastní zpracování dle 8, s. 24)

Relační datový model vychází z matematického pojetí relací a je složen z pěti hlavních složek, kterými jsou relace (tabulka), relační databáze, atribut (sloupec), doména atributu a datová n-tice (záznam) (4, s. 63).

V tomto datovém modelu představuje pojem relace tabulku, s řádky a sloupci, ve které jsou jednotlivé řádky tvořeny datovými n-ticemi a její sloupce tvoří atributy. Jednotlivé atributy mohou nabývat pouze hodnot, které jsou určeny doménou atributu. Relační databáze je tvořena jednotlivými relacemi (4, s. 63-65).

Jednotlivé relace mají následující vlastnosti:

- na pořadí jednotlivých řádků a sloupců nezáleží,
- neexistují dva stejné řádky a každý řádek odpovídá jedné n-tici,
- každá buňka obsahuje pouze jednu hodnotu,
- hodnoty v jednom sloupci jsou vždy stejného typu,
- názvy sloupců jsou jedinečné (7, s. 78).

1.5.4.1 Typy klíčů relačního datového modelu

V relačním datovém modelu můžeme definovat superklíč, kandidátní klíč, primární klíč a cizí klíč.

1.5.4.1.1 Superklíč

„Sloupec nebo množina sloupců, které jedinečně identifikují záznam v relaci.“ (4, s. 66)

Vzhledem k tomu, že superklíč může obsahovat i nadbytečné sloupce, je nutné identifikovat takové superklíče, které jsou složeny pouze ze sloupců nutných pro jednoznačnou identifikaci jednotlivých záznamů. Identifikací takovýchto superklíčů získáme možné kandidátní klíče (4, s. 66).

1.5.4.1.2 Kandidátní klíč

„Superklíč, který obsahuje jen minimální počet sloupců nutných k jedinečné identifikaci záznamů.“ (4, s. 66)

Jednotlivé kandidátní klíče se vyznačují dvěma vlastnostmi, a to že jsou jedinečné a neredukovatelné. Každá hodnota kandidátního klíče určuje pouze jeden konkrétní záznam a nelze jej již dále redukovat, aniž by došlo ke ztrátě jedinečnosti. Při volbě kandidátních klíčů je vždy nutné brát v potaz všechny hodnoty, kterých může kandidátní klíč nabývat, neboť skutečnost, že některý kandidátní klíč splňuje podmínku jedinečnosti ještě neznamená, že v budoucnosti nemůže nastat duplicita (4, s. 66 - 67).

1.5.4.1.3 Primární klíč

„Kandidátní klíč, který je vybrán, aby jedinečně určoval záznamy v tabulce.“ (4, s. 67)

Z jednotlivých kandidátních klíčů je vybrán primární klíč. Tento klíč slouží k jednoznačnému určení záznamu v tabulce a ostatní kandidátní klíče jsou označovány jako alternativní klíče. V extrémních případech může nastat situace, že jako primární klíč budou použity všechny sloupce tabulky (4, s. 67).

1.5.4.1.4 Cizí klíč

„Sloupec, nebo skupina sloupců v jedné tabulce, která odpovídá kandidátnímu klíči některé (případně téže) tabulky.“ (4, s. 67)

Cizí klíče slouží k propojení jednotlivých tabulek, kdy cizí klíč v jedné tabulce představuje primární klíč druhé tabulky. Názvy sloupců v obou tabulkách nemusí být shodné, ale musí se shodovat jejich doména (7, s. 86 - 87).

1.5.4.2 Integrita relačního modelu

Je třeba brát v potaz, že tvorba teoretického modelu dat vycházejícího z dat reálného světa s sebou nese určitá úskalí. Integritu modelu lze chápat jako situaci, kdy uložená data odpovídají vlastnostem jednotlivých objektů reálného světa. Integritní omezení lze rozdělit na omezení pro samotné entity a omezení pro jejich vztahy. Omezení pro entity lze dále rozdělit na doménovou, entitní a referenční integritu (8, s. 28).

1.5.4.2.1 Doménová integrita

Aby byla splněna podmínka doménové integrity, musí mít každý atribut definovaný datový typ a hodnoty, kterých může v rámci tohoto datového typu nabývat. Dále je nutné stanovit, zda bude zadání hodnoty vždy povinné, zda budou zadávané hodnoty jedinečné, případně zda budou jednotlivé hodnoty stanoveny číselníkem (6, s. 27 - 28).

1.5.4.2.2 Entitní integrita

Aby byla splněna podmínka entitní integrity, musí mít každá tabulka definovaný primární klíč, který nabývá pro každý záznam unikátních hodnot a díky tomu lze každý záznam v tabulce jednoznačně určit (6, s. 28).

1.5.4.2.3 Referenční integrita

Aby byla dodržena referenční integrita databáze, je nutné zamezit situaci, kdy bude existovat záznam s cizím klíčem, který neodpovídá hodnotě žádného primárního klíče tabulky v databázi. Zajištění referenční integrity lze docílit restriktivními pravidly pro práci se záznamy, které jsou uvedeny v následující tabulce (9, s. 24).

Tab. 1: Restriktivní pravidla pro zajištění referenční integrity
(Zdroj: 9, s. 25)

Akce	Hlavní tabulka	Související tabulka
Přidání záznamu	není omezeno	hodnota cizího klíče musí být v hlavní tabulce
Úprava záznamu	není možná, pokud se hodnota primárního klíče vyskytuje v související tabulce	hodnota cizího klíče musí být v hlavní tabulce
Odstranění záznamu	není možné, pokud se hodnota primárního klíče vyskytuje v související tabulce	neomezeně

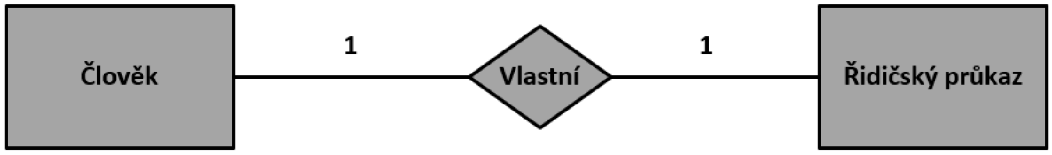
Alternativně je možné referenční integrity docílit pomocí kaskádových pravidel, při jejichž aplikaci dojde při změně primárního klíče v hlavní tabulce ke stejné změně u souvisejících hodnot v propojených tabulkách a při odstranění záznamu v hlavní tabulce dojde k odstranění všech záznamů propojených s odstraňovaným záznamem pomocí cizího klíče (9, s. 25).

1.5.4.2.4 Integritní omezení pro vztahy entit

Následující podkapitoly obsahují popis jednotlivých integritních omezení pro vztahy entit.

1.5.4.2.4.1 Vztah 1:1

Při tomto vztahu je jeden řádek jedné tabulky propojen s jedním řádkem z druhé tabulky. Tento vztah je poměrně ojedinělý, neboť je ve většině případů jednodušší takto propojené záznamy spojit do jedné tabulky. Nicméně může nastat situace, kdy počet potřebných sloupců v tabulce překročí maximální možné množství a v takovémto případě je nutné tabulky rozdělit na dvě se vztahem 1:1 (9, s. 23).

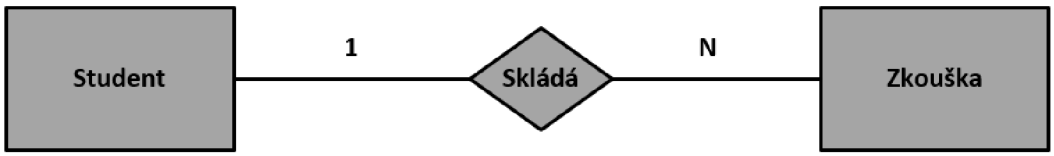


Obr. 8: Ukázka vztahu 1:1

(Zdroj: 6, s. 31)

1.5.4.2.4.2 Vztah 1:N

Jedná se o nejčastěji používaný vztah. Při tomto vztahu může jednomu řádku z první tabulky, která bývá označována jako hlavní, odpovídat žádný, jeden nebo více řádků v druhé tabulce označované jako podřízená, případně související. Jeden konkrétní záznam s primárním klíčem se může na druhé straně tohoto vztahu objevit jako cizí klíč vícekrát (9, s. 23).

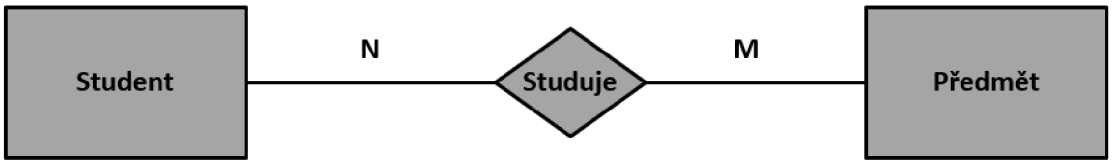


Obr. 9: Ukázka vztahu 1:N

(Zdroj: 6, s. 31)

1.5.4.2.4.3 Vztah N:M

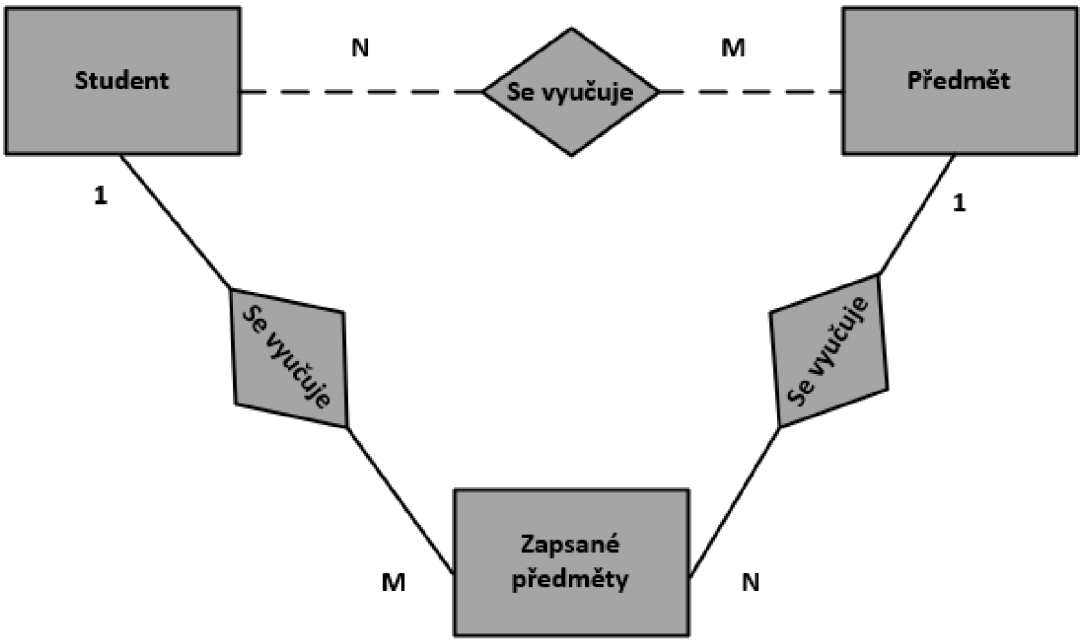
Při vztahu N:M odpovídá více záznamů z jedné tabulky více záznamům v druhé tabulce (9, s. 24).



Obr. 10: Ukázka vztahu N:M

(Zdroj: 6, s. 32)

Vzhledem k tomu, že takovýto vztah není možné realizovat přímo, je nutné vytvořit třetí tabulku, která slouží k propojení primárních klíčů z původních tabulek, kde existoval vztah N:M, čímž dojde ke vzniku dvou vztahů 1:N (9, s. 24).



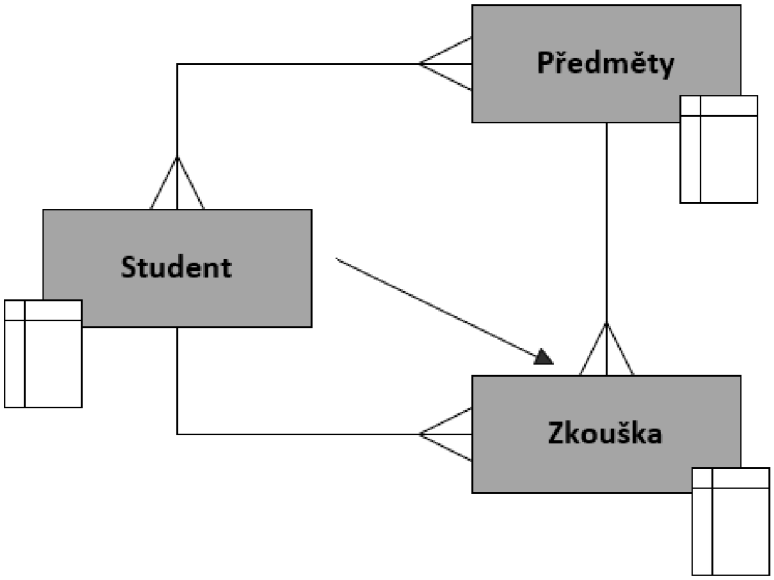
Obr. 11: Ukázka úpravy vztahu N:M

(Zdroj: 6, s. 34)

1.5.5 Objektový datový model

Objektový datový model je založen na jednotlivých objektech, které mají definované metody, které určují chování těchto objektů a vlastnosti, které tyto objekty popisují. S jednotlivými objekty lze pracovat pouze pomocí jejich metod (8, s. 24-25).

Každý objekt má přiřazený unikátní identifikátor, pomocí kterého lze mezi objekty vytvořit přímé vazby. Pokud jsou objekty stejného typu, lze hovořit o třídě objektů. Některá dostupná relační řešení byla postupně rozšířena o funkce spojené s objektovým modelem, v takovém případě hovoříme o objektově-relačních databázích (8, s. 24-25; 7, s. 451).



Obr. 12: Ukázka objektového datového modelu

(Zdroj: 8, s. 25)

1.6 Metodologie návrhu databáze

„Strukturovaný přístup používající procedury, techniky, nástroje a dokumentaci s cílem podpořit a usnadnit proces návrhu.“ (4, s. 206)

Metodologie návrhu databáze se skládá ze tří fází, jedná se o konceptuální, logický a fyzický návrh databáze. Rozdělení do jednotlivých fází slouží k udržení strukturovaného přístupu k návrhu databáze (4, s. 206).

1.6.1 Konceptuální návrh databáze

„Proces vytvoření modelu dat používaných v organizaci bez jakýchkoli úvah o fyzické implementaci.“ (4, s. 206)

Cílem této fáze návrhu databáze je vytvořit konceptuální model dat bez ohledu na fyzickou implementaci navrhované databáze. Konceptuální návrh se dělí do několika kroků a slouží jako podklad pro logický návrh. Jednotlivé kroky konceptuálního návrhu je možné vidět v následující tabulce (4, s. 206-208).

Tab. 2: Kroky konceptuálního návrhu databáze

(Zdroj: vlastní zpracování dle 4, s. 209-224)

Krok konceptuálního návrhu	Cíl daného kroku
Identifikace entit	identifikace hlavních objektů představujících jednotlivé entity
Identifikace relací	identifikace vztahů mezi jednotlivými entitami
Identifikace atributů	identifikace skutečností, kterými lze popsat entity a relace
Určení domén atributů	určení hodnot, kterých mohou jednotlivé atributy nabývat
Určení klíčových atributů	identifikace kandidátních klíčů, určení primárního klíče
Specializace/generalizace entit	volitelný krok, určení podtříd a nadtříd
Kontrola redundance	kontrola relací 1:1, odstranění redundantních relací
Kontrola podpory transakcí	návrh podporuje požadované transakce
Posouzení návrhu s uživateli	návrh splňuje požadavky uživatelů

1.6.2 Logický návrh databáze

„Proces vytvoření modelu dat používaných v organizaci, který je založen na specifickém modelu dat, ale nezávislý na konkrétním DBMS a jiných úvahách o fyzické implementaci.“ (4, s. 206)

Cílem této fáze návrhu databáze je převést výstupy z konceptuálního návrhu do podoby logického návrhu. V případě relačních databází se jedná o množinu relačních tabulek. Logický návrh se opět dělí do několika kroků a slouží jako podklad pro následující krok, kterým je fyzický návrh databáze (4, s. 206-208).

Tab. 3: Kroky logického návrhu databáze

(Zdroj: vlastní zpracování dle 4, s. 232-251)

Krok logického návrhu	Cíl daného kroku
Vytvoření tabulek	převedení výstupů z konceptuálního návrhu do množiny tabulek
Kontrola pomocí normalizace	navržené tabulky splňují pravidla normalizace
Kontrola podpory transakcí	návrh podporuje požadované transakce
Kontrola integritních omezení	návrh splňuje integritní omezení
Posouzení návrhu s uživateli	návrh splňuje požadavky uživatelů

1.6.3 Fyzický návrh databáze

„Proces vytvoření popisu implementace databáze ve vnější paměti; popisuje podkladové tabulky, organizaci souborů, indexy používané pro dosažení efektivního přístupu k datům, všechna související integritní omezení a bezpečnostní omezení.“ (4, s. 207).

Tato fáze návrhu databáze vychází z logického návrhu a opět se dělí do několika kroků. Vzhledem k tomu, že se jednotlivé DBMS značně liší je pro efektivní návrh databáze nutné detailně znát funkčnost vybraného DBMS (4, s. 258).

Tab. 4: Kroky fyzického návrhu databáze

(Zdroj:4, s.259-279)

Krok fyzického návrhu	Cíl daného kroku
Převod logického návrhu do cílového DBMS	vytvoření databáze na základě logického návrhu (návrh podkladových tabulek, reprezentace odvozených dat a zbývajících integritních omezení)
Volba organizace souborů a indexů	provedení analýzy transakcí, stanovení optimální organizace souborů a případné přidání indexů
Návrh uživatelských pohledů	navržení identifikovaných uživatelských pohledů
Návrh bezpečnostních mechanismů	navržení bezpečnostních mechanismů pro zajištění požadavků na bezpečnost
Zvážení zavedení redundance	navržení porušení pravidel normalizace v případě, že takovýto krok zlepší výkon
Monitorování a vyladění systému v provozu	úpravy, či opravy návrhu na základě sledování propustnosti transakcí, doby odezvy, či využití diskového prostoru

1.7 Proces normalizace

„Technika používaná pro vytvoření sady tabulek s minimální redundancí, která podporuje datové požadavky organizace.“ (4, s. 188)

Tato technika byla v roce 1972 vyvinuta Edgarem F. Coddem pro účely podpory návrhu relačních databází. Společným znakem normálních forem je, že jsou zaměřeny na vztahy jednotlivých sloupců v tabulkách. V praxi jsou nejvíce využívány první, druhá a třetí normální forma, nicméně normálních forem existuje více. Označení normálních forem a jejich posloupnost je následující:

- první normální forma (1NF),
- druhá normální forma (2NF),
- třetí normální forma (3NF),
- Boyce-Coddova normální forma (BCNF),
- čtvrtá normální forma (4NF),
- pátá normální forma (5NF) (4, s. 188; 10).

1.7.1 První normální forma

První normální forma řeší zejména odstranění vícehodnotových atributů. Tato normální forma je kritická pro vytvoření vhodné struktury jednotlivých tabulek relační databáze. Aby byla splněna podmínka této normální formy, musí každý sloupec tabulky pro daný záznam obsahovat pouze jedinou hodnotu (4, s. 191; 11, s. 80).

1.7.2 Druhá normální forma

„Ve druhé normální formě (2NF) je relace, když je v 1NF a každý neklíčový atribut je plně funkčně závislý na primárním klíči relace.“ (11, s. 80)

Tato normální forma řeší funkční závislost mezi jednotlivými atributy. Pokud má návrh tabulky splňovat tuto formu, musí splňovat podmínky první normální formy a zároveň všechny neklíčové atributy jsou plně závislé na celém primárním klíči (4, s. 192-193).

Pokud je primární klíč tvořen pouze jedním atributem, pak tabulka rovnou splňuje tuto normální formu (4, s. 192).

1.7.3 Třetí normální forma

„Ve třetí normální formě je relace, jestliže je v 2NF a každý neklíčový atribut je netranzitivně závislý na primárním klíči.“ (11, s. 81)

Pro splnění této normální formy je opět nutné, aby tabulka splňovala podmínky předchozí normální formy a zároveň musí být všechny neklíčové atributy vzájemně nezávislé (4, s. 195).

1.7.4 Boyce-Coddova normální forma

„Relace je v Boyce-Coddově normální formě právě tehdy, jestliže každý determinant funkční závislosti v relaci je zároveň kandidátním klíčem.“ (11, s. 81)

Boyce-Coddova normální forma je přísnější variantou třetí normální formy. Pokud relace splňuje podmínky této normální formy je zároveň ve všech předchozích normálních formách. Nicméně pokud je relace ve třetí normální formě neznamená to, že splňuje podmínky této normální formy (6, s. 59).

Pokud platí, že relace má dva nebo více kandidátních klíčů, z nichž jsou alespoň dva složené a některé jejich atributy se vzájemně překrývají, pak je relace v této normální formě právě tehdy, když mezi jednotlivými kandidátními klíči neexistuje funkční závislost (6, s. 59-60).

1.7.5 Čtvrtá normální forma

„Ve čtvrté normální formě (4NF) je relace tehdy, je-li v BCNF a všechny vícehodnotové závislosti obsažené v relaci jsou zároveň funkčními závislostmi.“ (11, s. 82)

1.7.6 Pátá normální forma

Relace je v páté normální formě, pokud splňuje podmínky předchozí normální formy a nejde ji již dále rozkládat bez ztráty informace (11, s. 83).

1.8 Jazyk SQL

Jazyk SQL (Structured Query Language) je nejvíce využívaným databázovým jazykem. Jedná se o textově orientovaný deklarativní jazyk vytvořený pro přístup k relačním databázím (12, s. 33).

Prvotní vývoj tohoto jazyka probíhal v laboratořích IBM mezi lety 1974–79 v rámci projektů SEQUEL a System/R. Následně byl jazyk SEQUEL změněn na SQL, neboť bylo zjištěno, že slovo SEQUEL má jako ochranou známku registrovanou společnost Hawker-Siddeley Aircraft Company z Velké Británie. V roce 1986, respektive 1987, vznikly komise pro standardizaci tohoto jazyka při institutu ANSI (American National Standarts Institute) a následně při organizaci ISO (International Organization for Standardization) (12, s. 37-38).

Příkazy jazyka SQL se svojí strukturou podobají anglickým větám, každý příkaz začíná klíčovým slovem a je ukončen oddělovačem (některá řešení SQL koncový oddělovač nepožadují), kterým je ve většině případů středník. Jednotlivé části těchto příkazů se oddělují mezerami, na jejichž počtu nezáleží. Této skutečnosti lze využít pro větší přehlednost zápisu, přičemž ale musí být dodržena posloupnost jednotlivých částí. Položky se oddělují čárkou, za kterou nemusí následovat mezera, případně může být mezer více pro větší přehlednost zápisu (12, s. 39-40; 9, s. 29).

Názvy mohou být zapsány pouze velkými, či malými písmeny (některá řešení velká a malá písmena mohou rozlišovat), čísly a pro větší přehlednost lze využít znak podtržítka pro oddělení jednotlivých slov, ze kterých je název složen. Jednotlivé názvy se zapisují bez apostrofů a uvozovek a pro jejich tvorbu nelze využít slova, která mají v rámci SQL dotazování určený význam. (12, s. 39-40; 9, s. 29).

Při zápisu textových řetězců je nutné tyto řetězce ohraničit apostrofy, případně uvozovkami. Na rozdíl od textových řetězců se čísla apostrofy, či uvozovkami nikdy neohraničují a pro oddělení desetinné části slouží tečka (12, s. 39-40; 9, s. 29).

Začátek jednořádkového komentáře začíná dvěma pomlčkami (-- Jednořádkový komentář.), více řádkový komentář začíná lomítkem a hvězdičkou a je ukončen hvězdičkou a lomítkem (/* Víceřádkový komentář. */). Zápis datumů je obvykle ohraničen apostrofy a je zapsán ve formátu RRRR-MM-DD (12, s. 39-40; 9, s. 29).

Jednotlivé příkazy jazyka SQL lze na základě jejich funkce rozdělit do pěti samostatných kategorií, ze kterých všechny mají stejnou syntaxi a pravidla. Jedná se o tyto kategorie:

- DDL (Data Definition Language),
- DQL (Data Query Language),
- DML (Data Manipulation Language),
- DCL (Data Control Language),
- příkazy řízení transakcí (12, s. 40).

Příkazy jazyka DDL umožňují vytvářet jednotlivé databázové objekty (klíčové slovo CREATE), upravovat jednotlivé databázové objekty (klíčové slovo ALTER) a mazat databázové objekty (klíčové slovo DROP) (12, s. 40).

Pomocí příkazů jazyka DQL lze načítat data z databáze (klíčové slovo SELECT) (12, s. 41).

Příkazy jazyka DML slouží k práci s daty uloženými v databázi. Data lze do databáze vkládat (klíčové slovo INSERT), upravovat (klíčové slovo UPDATE) a případně mazat (klíčové slovo DELETE) (12, s. 41).

Příkazy jazyka DCL slouží k nastavení přístupu k datům uloženým v databázi (klíčová slova GRANT a ALTER) (12, s. 41).

Příkazy řízení transakcí slouží k tvorbě nedělitelné sady příkazů, která se buď provede úspěšně celá, nebo se neprovede vůbec, a to nezávisle na ostatních existujících transakcích. Transakce převádí databázi z jednoho stavu do druhého a tato změna je trvalá, a to i v případě chyby, či vypnutí databáze (klíčová slova BEGIN, COMMIT a ROLLBACK) (12, s. 41, 161).

1.9 Datové typy

Základní datové typy, které je potřeba dále podrobněji členit, lze obecně rozdělit na:

- číselné datové typy,
- textové datové typy,
- časové datové typy,
- a logické datové typy (9, s. 16; 13).

Následující tabulka zobrazuje podrobnější členění vybraných datových typů používaných v MySQL.

Tab. 5: Vybrané datové typy MySQL
(Zdroj: vlastní zpracování dle 18, s. 224; 13)

Název datového typu	Typ	Rozsah
TINYINT	celé číslo	0 až 255
SMALLINT		0 až 65 535
MEDIUMINT		0 až 16 777 215
INT		0 až 4 294 967 255
BIGINT		0 až 18 446 744 073 709 551 615
FLOAT	čísla s pohyblivou desetinnou čárkou	±1,175494351E−38
DOUBLE		±2,2250738585072014E-308
DECIMAL		stejný jako u DOUBLE
DATE	datum	01.01.1000 do 31.12.9999
DATETIME	datum a čas	stejný jako u DATE
TIMESTAMP	časové razítko	1.1.1970 až 2037
TIME	čas	-838:59:59 až 839:59:59
YEAR (2/4)	rok	1901 až 2155 (4) nebo 1970 do 2069 (2)
CHAR	text, pevná délka	0 až 255
VARCHAR	text, plovoucí délka	0 až 255

Logické hodnoty jsou v rámci MySQL reprezentovány jako TINYINT (1), kde hodnota 0 reprezentuje logickou hodnotu False, a jakákoliv jiná přípustná hodnota reprezentuje logickou hodnotu True (14).

1.10 ER diagram

ER diagram slouží ke grafické reprezentaci datového modelu a vychází z něho všechny současné reprezentace datových modelů. Poprvé ER diagram prezentoval Peter Chen v roce 1976, následně byl tento model v roce 1988 přijat za standard organizací ANSI (11, s. 9-10, 41).

Pro tvorbu ER diagramů lze využít jeden z následujících stylů:

- Chenův styl,
- Bachmanův styl,
- Martinův styl,
- inženýrský styl,
- zjednodušený styl (6, s. 49-51).

Obr. 13: Ukázka zjednodušeného stylu

(Zdroj: 6, s. 51)

1.11 XAMPP

XAMPP je open source řešení vyvinuté v rámci neziskového projektu Apache Friends. Cílem tohoto projektu je vytvořit snadno instalovatelný balíček zahrnující webový server Apache, databázi MariaDB (MySQL) a programovací jazyky PHP a Perl (15; 16).

Název XAMPP je akronymem tvořeným z počátečních písmen slov popisujících tento balíček. Jedná se o:

- X – cross-platform (multiplatformní),
- A – Apache,
- M – MySQL,
- P – PHP,
- P – Perl (17).

Vzhledem k nízkému zabezpečení tohoto řešení je vhodné XAMPP používat pouze jako vývojové prostředí, nikoliv pro ostrý provoz (18, s. 216).

2 ANALÝZA SOUČASNÉHO STAVU

Tato kapitola obsahuje analýzu potřeb drobných chovatelů v souvislosti s evidencí dat spojených s chovem ovcí, koz, psů a drůbeže. Tito chovatelé se zabývají zájmovým chovem a evidované údaje jsou přizpůsobeny skutečnosti, že cílem těchto chovatelů není tvorba zisku, ale potěšení z chovatelské činnosti.

2.1 Stávající způsob evidence

V současné době probíhá zaznamenávání pouze některých informací, které by chtěli drobní chovatelé evidovat. Jedná se zejména o zjednodušenou evidenci finančních transakcí spojených s chovem, která neumožnuje strukturované vyhodnocování, a evidenci produktů získaných vlastní činností (mléko, vejce), které jsou hlavním výnosem chovatelské činnosti těchto chovatelů.

Tato evidence byla dříve vedena v čistě papírové formě, v současnosti někteří chovatelé využívají pro evidenci kancelářský software, například Excel. Tento software jim umožňuje alespoň základní výpočty spojené s evidovanými daty.

Drobní chovatelé mají většinou k dispozici pouze osobní počítač s připojením k internetu, případně mobilní zařízení (mobilní telefon, tablet). Co se týče používání informačních technologií, mají drobní chovatelé většinou pouze povrchní uživatelskou znalost.

2.2 Chov ovcí a koz

Pokud chovatel chová jeden a více kusů ovcí nebo koz, musí být takovýto chov realizován prostřednictvím registrovaného hospodářství. V rámci hospodářství je evidován chovatel, jednotlivé druhy chovaných zvířat a místo, kde jsou tato zvířata chována.

Každá ovce nebo koza má přiděleno registrační číslo, díky kterému lze konkrétního jedince přesně identifikovat. Označení ovcí koz se provádí pomocí trvalých ušních známek, které jsou aplikovány pomocí speciálních kleští. Každý jedinec musí být označen nejpozději do šesti měsíců věku nebo do doby, než opustí hospodářství.

Ovce jsou společně s beranem chovány celoročně na pastvinách s přístupem do krytého přístřešku, ve kterém mají přístup k vodě, lizu a senu.

K oplodnění dochází na podzim a průměrná délka březosti ovce je přibližně 150 dní, nicméně vzhledem ke skutečnost, že jsou ovce chovány společně s beranem nelze přesně odhadnout datum porodu. Již krátce po narození jehňat jsou jehňata společně s ovcí vyháněna na pastvinu, kde si mají možnost postupně přivyknout na příjem trávy, případně sena.

Kozy jsou chovány ve chlévě a v době růstu trávy jsou každé ráno vyháněny na pastvu, kde jsou po celý den, a následně jsou ve večerních hodinách opět přehnány do chléva. Jak ve chlévě, tak i na pastvinách mají kozy neustále přístup k vodě, lizu, senu a případně dalším krmivům.

Délka březosti trvá stejně jako u ovcí přibližně 150 dní a k oplodnění dochází na podzim. Kůzlata během růstu postupně začínají přijímat pevná krmiva, až je příjem pevných krmiv dostatečný, je zahájeno dojení koz. Mléko je dále zpracováváno pro vlastní potřebu.

V jednotlivých výbězích a chlévech jsou nainstalována různá chovatelská zařízení, jakými mohou být například napáječky, krmítka, elektrické ohradníky a jiné. Vzhledem k tomu, že i v drobném chovu může být takovýchto zařízení poměrně velké množství, mohou znamenat poměrně významnou finanční položku, kterou je třeba vynaložit na chov.

Drobní chovatelé v rámci chovu ovcí a koz potřebují evidovat následující informace:

- registrační číslo jedince,
- jméno ovce/kozy,
- datum narození,
- plemeno,
- pohlaví,
- označení, zda je daný jedinec chovný,
- vlastnosti,
- porážkovou hmotnost,

- narozená jehňata a kůzlata,
- umístění jedinců,
- kontaktní informace na původního majitele,
- absolvování případných chovatelských akcí,
- evidence dojení,
- u ovcí evidence stříhání vlny,
- evidence veterinárních zákroků.

V rámci zájmového chovu může chovatel vyhodnocovat například následující ukazatele:

- porodnost,
- úmrtnost mláďat a dospělých jedinců
- a dojnost.

2.3 Chov psů

Pro chov psů s průkazem původu je nutné mít registrovanou chovatelskou stanici s mezinárodně chráněným názvem. Chov psů je regulován zákony a vyhláškami, nicméně chovatelské kluby mají většinou podmínky chovu oproti zákonným minimům ještě zpřísněny.

Průkaz původu mohou v České republice pod hlavičkou FCI vydávat dvě instituce, konkrétně se jedná o Českomoravskou kynologickou unii a Český kynologický svaz.

Hlavním identifikátorem průkazu původu je číslo zápisu, které se skládá z identifikace plemenné knihy, zkratky názvu plemene, pořadí jedince, roku narození, případně roku uchovnění. Číslo zápisu může například vypadat následovně: CMKU/BSB/XXX1/17/20.

Samotné uchovnění jedince může být podmíněno absolvováním svodu mladých, bonitace, výstavy, případně splnění dalších podmínek (rentgen kyčlí aj.). Konkrétní podmínky pro uchovnění jsou závislé na požadavcích stanovených daným klubem plemene.

Pro chov je v některých případech nutné zajistit vystavení krycího listu. Žádost o vystavení krycího listu zasílá majitel chovné feny poradci chovu. Krycí list obsahuje označení feny a vybraných plemeníků. Krycí list je potvrzení, že daná fena může být nakryta konkrétním psem (plemeníkem). Po provedení krytí i případném narození mláďat je třeba krycí list doplněný o příslušné informace zaslat poradci chovu.

V průběhu březosti je sledován zdravotní stav feny, jsou jí upravovány krmné dávky adekvátně k aktuální kondici a probíhají různé veterinární kontroly (zpravidla vyšetření ultrazvukem pro potvrzení březosti a zjištění počtu štěňat).

Po narození štěňat je pro každé štěně nutné zaznamenat čas narození, porodní váhu, pohlaví, barvu srsti, paspárky a barevné označení štěněte. V prvních dnech po porodu probíhá kontrola veterinářem, pravidelné vážení, sledování přírůstků štěňat a jejich odčervení. Zároveň je nutné provést hlášení o vrhu poradci chovu, který následně v určeném intervalu provádí kontroly vrhu.

Ve dvou až třech týdnech věku štěňat může chovatel zažádat plemennou knihu o vydání tetovacích čísel. Kolem třetího týdne věku dochází k odstavení štěňat od matky, po 42. dnu věku štěněte může být provedeno čipování, případně ještě tetování štěňat, a další očkování. Mezi 6. a 7. týdnem proběhne kontrola štěňat oblastním poradcem chovu, během které je provedena kontrola označení, váhy štěňat a identifikace feny. Odběr štěněte je možný po 50. dnu od narození.

Drobní chovatelé v rámci chovu psů potřebují evidovat následující informace:

- číslo zápisu,
- registrované jméno jedince,
- volací jméno jedince,
- pohlaví,
- datum a čas narození,
- číslo čipu a tetování,
- vlastnosti,
- kontaktní informace na chovatele,

- kontaktní informace na majitele,
- evidence výšky a váhy,
- plánování aktivit (výcviku),
- absolvování případných chovatelských akcí,
- seznam krycích listů,
- evidence veterinárních zákroků.

V rámci zájmového chovu může chovatel vyhodnocovat následující ukazatele:

- počet štěňat na vrh,
- vývoj růstu v porovnání s tabulkovou hodnotou.

2.4 Chov drůbeže

Pro chov drůbeže vzniká povinnost registrace, pokud:

- chovatel chová více jak 100 ks drůbeže, která produkuje násadová vejce,
- chovatel chová více než 500 kusů drůbeže,
- chovatel produkuje vejce určená k prodeji,
- nebo provozuje líheň, která má kapacitu 100 a více kusů násadových vajec.

Značení jedinců je realizováno prostřednictvím barevných rozlišovacích kroužků, případně pokud je chovatel členem Českého svazu chovatelů pomocí registračních kroužků s číslem.

Samci a samice drůbeže jsou společně chováni v jednotlivých výbězích. Jednotlivé výběhy jsou vybaveny krmítky a napáječkami. Součástí výběhů jsou i zařízení pro uzavření jedinců přes noc z důvodu bezpečnosti.

Vejce jsou v době snůšky každý den sbírána a jsou následně využita. V případě násadových vajec jsou postupně sbírána, skladována a následně vložena do líhně. Tento způsob většina chovatelů volí, protože u velkého množství druhů drůbeže byla šlechtěním potlačena kvokavost.

Doba líhnutí se u jednotlivých druhů drůbeže liší, například líhnutí ze slepičích vajec trvá 21 dní, během kterých je nutné vejce dvakrát prosvítit. Prosvěcování slouží k potvrzení správného vývoje plodu. Pokud vejce projde kontrolou, je ponecháno v líhni, v opačném případě je vyřazeno.

Drobní chovatelé v rámci chovu drůbeže potřebují evidovat následující informace:

- číslo kroužku,
- datum vylíhnutí,
- plemeno drůbeže,
- pohlaví drůbeže,
- porážkovou hmotnost,
- líhnutí vajec,
- umístění jedinců,
- absolvování případných chovatelských akcí,
- evidenci veterinárních zákroků.

2.5 Evidence nákladů a výnosů spojených s chovem

Prvotní náklady na chov zvířat jsou spojené s nákupem nových jedinců a zajištění vhodného prostředí pro jejich chov (výběhy, kotce, ohrady, chlévy a jiné). Mimo jiné je chov zvířat spojený s nákupem relativně velkého množství pomůcek k chovu, při chovu ovcí a koz se může jednat například o elektrické ohradníky, krmítka, či napáječky. Při chovu psů je možné pořizovat různé pamlsky, hračky, pelíšky a jiné.

Dále je chov spojen s náklady na veterinární péči (pravidelné očkování, léčení a jiné) a krmiva (granule, směsi, doplňky krmiv a jiné). Dále jsou hrazeny různé poplatky za výstavy a chovatelské akce. U psů lze též brát v potaz náklady spojené s výcvikem a případnými volnočasovými aktivitami.

Výnosy z chovu jsou spojeny s prodejem chovaných zvířat, či vlastní spotřebou produktů vzniklých chovem hospodářských zvířat.

2.6 Krmení

Drobní chovatelé při větším počtu chovaných zvířat potřebují mít přehled o tom, na jak dlouho jim vystačí zásoby krmiv. Z tohoto důvodu je třeba krmení plánovat vzhledem k počtu chovaných jedinců. Krmení probíhá na základě aktuálních potřeb zvířat (březost, roční období a jiné).

V některých případech může být zvíře krmeno jednou denně, ale může nastat i situace kdy bude potřeba krmit vícekrát za den. Například dospělé psy lze krmit jednou denně, štěňata jsou krmena častěji. Hospodářská zvířata, která jsou chována společně, nejsou krmena jednotlivě, ale jsou krmena společně v rámci lokace, ve které jsou chována.

Jednotlivá krmiva mohou být určena pro více druhů zvířat nebo pro konkrétní druh jako jsou například granule pro psa. Tato krmiva se mohou lišit například složením, či stářím jedince, pro kterého jsou určena.

2.7 Evidence zásob

Drobní chovatelé též potřebují evidovat zásoby, jejich umístění, množství včetně měrné jednotky, transfery mezi jednotlivými umístěními a vývoj těchto veličin v čase.

Zásoby drobných chovatelů jsou složeny z následujících položek:

- zásoby krmiv,
- zásoby chovatelských pomůcek,
- zásoby vzniklé vlastní činností (vejce, mléko, vlna a jiné).

2.8 Shrnutí požadavků na databázi

Z předchozích kapitol vyplývají základní požadavky drobných chovatelů. Mezi hlavní požadavky na evidenci patří:

- evidence chovaných zvířat a základních informací o nich,
- evidence produktů získaných samotným chovem,
- evidence krmiv a plánování krmení,

- evidence nákladů a výnosů spojených s chovem,
- evidence celkových zásob spojených s chovem.

Vzhledem ke skutečnosti, že drobní chovatelé nedisponují rozsáhlou výpočetní technikou, ve většině případů mají k dispozici pouze počítač s připojením k internetu, je potřeba navrhované řešení této skutečnosti uzpůsobit.

2.9 Dostupné platformy

Následující podkapitoly obsahují stručný přehled nejpopulárnějších platforem, ze kterého bude následně vybrána platforma pro realizaci navrhovaného řešení.

2.9.1 Oracle

Vývoj tohoto řešení probíhá od 70. let a v současná době je dostupné velké množství edic, které pokrývají nejrůznější potřeby organizací. Nejnovější verze podporuje i cloudové řešení, které může být hostované na jednom, či více serverech. Hlavní výhodou tohoto řešení je, že je robustní, je neustále inovováno a dokáže pracovat s extrémně velkými objemy záznamů. Nevýhodou tohoto řešení jeho vysoká cena, které ho činní pro malé organizace nedostupným, a jeho vysoké nároky na zdroje (19).

2.9.2 MySQL

Jedná se o nejpopulárnější databázi pro webové aplikace. MySQL je dostupné jak jako freeware, tak i v placené verzi pro komerční užití. Toto řešení je spíše orientované na rychlost a spolehlivost, než na množství dostupných funkcí. Řešení má jednoduché uživatelské rozhraní a umožňuje zpracování velkých objemů dat. Výhodou tohoto řešení je možnost spolupráce s jinými řešeními například Oracle. Nevýhodou řešení je, že oproti jiným neobsahuje některé automatické procesy, není v něm zabudována podpora pro XML a OLAP a podpora k free verzi je placená (19).

2.9.3 Microsoft SQL Server

Toto řešení má dostupné velké množství edic a lze jej využít v cloudovém prostředí stejně tak dobře jako lokálně. V současné době je k dispozici pro platformu Windows i platformu Linux. Mezi výhody řešení patří, že umožňuje sledování vývoje dat v čase, či řízení přístupu k citlivým datům. Jedná se o rychlý a stabilní systém umožňující řízení výkonu za účelem optimalizace využití zdrojů, který je vhodný k propojení s ostatními produkty od společnosti Microsoft. Značnou nevýhodou je cena, která toto řešení v nejvyšší edici činí pro velké množství organizací nedostupným (19).

2.9.4 PostgreSQL

Jedná se o volně dostupné řešení, které je často využívané pro webové databáze. Umožňuje uživatelům zpracovávat jak strukturovaná, tak i nestrukturovaná data. Lze jej použít napříč jednotlivými platformami včetně Linuxu. Lze jej využít ve virtuálním, fyzickém, či cloudovém prostředí. Poslední verze umožňuje práci s velkými objemy dat. V současné době probíhá vylepšení zabezpečení. Výhodou je, že je škálovatelné a zvládne pracovat s extrémně velkými objemy dat. Další výhodou je podpora práce s JSON, velké množství předdefinovaných funkcí a množství dostupných rozhraní. Velkou nevýhodou je nekompletnost dostupné dokumentace (19).

2.9.5 MongoDB

Toto řešení je navržené pro práci se strukturovanými i nestrukturovanými daty a je dostupné jak ve free verzi, tak i ve verzi pro komerční užití. Pro připojení databáze k aplikaci používá takzvané ovladače, kterých je v současné době dostupná celá řada. Řešení nebylo vyvinuto k práci s relačním datovým modelem, proto při práci s tímto modelem může docházet k výkonnostním problémům. Výhodou je rychlost, snadné používání, podpora JSON a dalších noSQL dokumentů. Toto řešení také umožňuje ukládat a zpracovávat data v libovolné struktuře (19).

Nevýhodu je, že jako dotazovací jazyk není používán jazyk SQL, nastavení je poměrně komplikované a v případě využití výchozího nastavení může docházet k problémům se zabezpečením (19).

2.9.6 MariaDB

Jedná se open-source řešení, které je dostupné jak ve free, tak i v placené verzi. Toto řešení je velice podobné MySQL, neboť někteří vývojáři se podíleli na vývoji obou těchto řešení. V rámci tohoto řešení je optimalizován dotazovací výkon. Výhodou je rychlost a stabilita, otevřenost a dostupnost šifrování jak na úrovni síťové, serverové, tak i aplikační. Nevýhodou je, že se jedná poměrně nové řešení a stejně jako v případě jiných řešení je jeho podpora placená (19).

2.9.7 DB2

Jedná se o řešení, které dokáže pracovat s noSQL, kromě toho umí pracovat s JSON a XML soubory. Jedná se o řešení, které je vytvořeno pro práci na serverech IBM a verze pro pracovní stanice podporuje Windows, Unix a Linux. Díky využití technologie přeskakování dat je docíleno zvýšení rychlosti. Toto řešení obsahuje vylepšené nástroje pro obnovu, kompatibilitu, či analýzy. Výhodami tohoto řešení jsou možnost využití pro enormně velké databáze, možnost nasazení v cloudu i lokálně a souběžné spuštění více úloh pomocí plánovače. Mezi nevýhody lze zařadit poměrně vysoké náklady pro malé organizace, některé funkce vyžadují software třetích stran a po třech letech je podpora dostupná pouze za příplatek (19).

2.9.8 SAP HANA

Toto řešení je sloupcově orientované a umí pracovat jak se SAP, tak i non-SAP daty. Je navrženo tak, aby mohlo ukládat a získávat data z různých aplikací a dalších zdrojů napříč různými vrstvami. Stejně tak jako jiná řešení, může být použito jak lokálně, tak i v cloudu. Výhodami jsou podpora SQL, OLTP a OLAP, snižování požadavků na zdroje, ukládání dat do paměti, což vede ke snížení času potřebnému k přístupu k těmto datům, umožnění reportingu, management zásob v reálném čase a možnost propojení s velkým množstvím aplikací. Podstatnou nevýhodou je, že ve srovnání s ostatními řešeními se jedná o poměrně drahou variantu. Další nevýhodou jsou časté změny kódu (19).

2.10 Výběr platformy

V předchozí podkapitole jsem analyzoval dostupné platformy a popsal jejich výhody a nevýhody. Při výběru platformy jsem zvažoval následující parametry:

- finanční náročnost na pořízení databáze,
- dostupnost prostředí pro vývoj,
- objem dat, pro který je databáze určena,
- možnosti napojení na uživatelské rozhraní.

Z výše uvedených parametrů z výběru vypadla řešení určená pro velké a střední podniky, která jsou pro drobné chovatele ekonomicky nedostupná.

Při hledání vhodného prostředí, v jehož rámci by bylo možné navrhované řešení realizovat, jsem objevil prostředí XAMPP, které obsahuje databázi MariaDB. Toto prostředí je pro vývoj navrhovaného řešení vhodné zejména proto, že databáze v něm obsažená, je vhodná pro práci s očekávaným objemem dat a splňuje i ostatní uvedené požadavky, a zároveň toto prostředí umožňuje propojit vývoj samotné databáze s vytvořením uživatelského rozhraní v jednotném prostředí.

3 VLASTNÍ NÁVRHY ŘEŠENÍ

Tato kapitola obsahuje jednotlivé části návrhu databáze, jedná se o konceptuální, logický a fyzický návrh databáze.

3.1 Základní vymezení navrhovaného řešení

V rámci tohoto řešení je předpokládaný počet uživatelů v řádu jednotek. Dalším předpokladem je, že každý uživatel bude mít k dispozici jeden přístup a pod tímto přístupem budou prováděny veškeré aktivity.

Požadavky drobných chovatelů byly popsány v předchozí kapitole. V rámci této práce je zpracován samotný návrh databáze, nicméně dalším krokem bude vytvoření webového rozhraní pro práci s touto databází. Z tohoto důvodu byla databáze vytvořena prostřednictvím prostředí XAMPP, které dále umožňuje při vytváření webového rozhraní jednoduše navázat na vytvořenou databázi.

Vzhledem k tomu, že se skupiny chovaných zvířat jednotlivými chovateli liší a každá skupina chovaných zvířat se vyznačuje drobnými odlišnostmi na požadovanou evidenci, je třeba aplikaci dělit do jednotlivých modulů, z nichž si budou moci uživatelé vybírat pouze moduly, které chtějí využívat, čímž dojde k odstranění nepotřebných funkcionalit z uživatelského rozhraní. V současné době, řešení počítá se čtyřmi moduly, konkrétně se jedná o:

- ovce,
- kozy,
- psi
- a drůbež.

Vhledem k tomu, že je řešení navrženo pro využití více uživateli, je návrh koncipován tak, aby umožnil jednotlivým uživatelům jejich vlastní nastavení. Jedná se zejména o číselníky, na které bude později navázáno analytické vyhodnocování.

3.2 Konceptuální návrh databáze

Tato kapitola obsahuje identifikaci jednotlivých entit spojených se zájmových chovem zvířat a identifikaci jejich vzájemných vztahů.

3.2.1 Identifikace základních entit

V rámci této kapitoly jsou identifikovány základní entity. Tyto entity představují objekty, se kterými chovatel v reálném světě pracuje. Dále jsou uvedeny jejich popisy, možné alternativní označení a jejich předpokládaný počet výskytů za očekávané množství uživatelů. Tyto informace jsou shrnuty v následující tabulce.

Tab. 6: Identifikace základních entit

(Zdroj: vlastní zpracování)

Název entity	Popis entity	Alias entity (alternativní označení)	Předpokládaný počet výskytů
Chovatel	subjekt zabývající se zájmovým chovem	uživatel	jednotky
Zvíře	předmět zájmového chovu		stovky
Druh zvířete	označení druhu chovaného zvířete (ovce, koza, pes a jiné)		desítky
Plemeno zvířete	označení plemene chovaného zvířete (Šumavská ovce, Romanovská ovce a jiné)		stovky
Vlastnosti zvířete	exteriérové, povahové a jiné vlastnosti zvířat		stovky
Umístění	evidence míst, na kterých jsou zvířata chována, případně kde je uchováván materiál, či krmiva	lokace	stovky
Typ umístění	členění umístění do jednotlivých skupin dle potřeb uživatele	typ lokace	desítky
Růstová tabulka	evidence výšky a váhy zvířete v čase		tisíce
Aktivity zvířete	evidence jednotlivých aktivit (výcvik)		tisíce

Chovatelské pomůcky	evidence pomůcek, které jsou používány k chovu	materiál	tisíce
Chovatelské akce	výstavy a zkoušky zvířat		stovky
Chovatelské stanice	evidence chovatelský stanic chovatele		desítky
Narození mláďat	evidence narozených mláďat		tisíce
Líhnutí drůbeže	evidence líhnutí vajec		tisíce
Krycí list	evidence vytavených krycích listů		stovky
Dojení zvířat	evidence provedení dojení zvířat		tisíce
Stříhání zvířat	evidence provedení stříhání vlny zvířat		stovky
Veterinář	subjekt, který provádí veterinární ošetření	zvěrolékař	stovky
Veterinární ošetření	evidence provedených veterinárních ošetření		tisíce
Krmivo	evidence krmiv používaných k chovu		stovky
Zásoby	evidence zásob, které má chovatel k dispozici		tisíce
Finanční operace	evidence nákladů a výnosů spojených s chovem		tisíce

3.2.2 Identifikace základních vztahů mezi entitami

Tato kapitola představuje identifikaci základních vztahů mezi entitami, které byly představeny v předchozí kapitole. Tyto vazby vychází z vazeb, které mezi těmito entitami existují v reálném světě. Jak je z jejich soupisu patrné, existuje v reálném světě mezi některými entitami vazeb více. Příkladem může být dvojice entit chovatel a zvíře, kdy chovatel jednak provádí evidenci chovaných zvířat, jednak provádí různé úkony spojené s chovem (dojení, stříhání vlny a jiné).

Následující tabulka zobrazuje tyto vztahy.

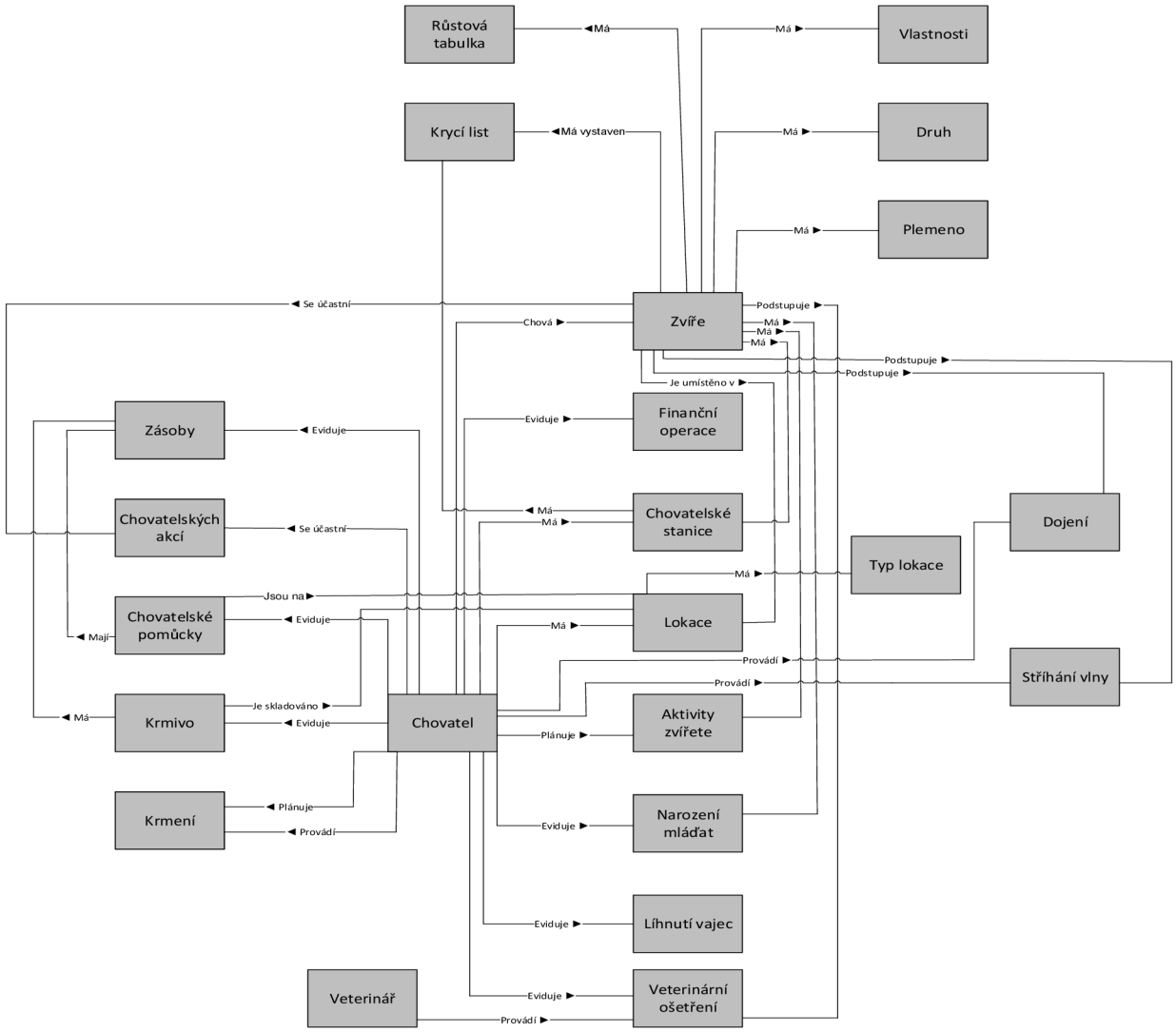
Tab. 7: Identifikace základních vztahů

(Zdroj: vlastní zpracování)

Název entity	Vztah	Název entity
Chovatel	chová	zvíře
Chovatel	má	lokace
Chovatel	má	chovatelské stanice
Chovatel	eviduje	krmivo
Chovatel	eviduje	chovatelské pomůcky
Chovatel	se účastní	chovatelských akcí
Chovatel	eviduje	narození mláďat
Chovatel	eviduje	líhnutí vajec
Chovatel	eviduje	finanční operace
Chovatel	eviduje	zásoby
Chovatel	eviduje	veterinární ošetření
Chovatel	plánuje	krmení
Chovatel	provádí	krmení
Chovatel	provádí	dojení
Chovatel	plánuje	aktivity zvířete
Chovatel	provádí	stříhání vlny
Zvíře	má	druh
Zvíře	má	plemeno
Zvíře	má	vlastnosti
Zvíře	podstupuje	veterinární ošetření
Zvíře	se účastní	chovatelských akcí
Zvíře	má	aktivity zvířete
Zvíře	je umístěno v	lokaci
Zvíře	má	růstovou tabulku
Zvíře	má	chovatelskou stanici
Zvíře	má	mláďata
Zvíře	má vystaven	krycí list
Zvíře	podstupuje	dojení
Zvířeti	podstupuje	stříhání
Umístění	má	typ
Chovatelské pomůcky	jsou na	lokaci
Chovatelské pomůcky	mají	zásobu
Chovatelská stanice	má	krycí listy
Veterinář	provádí	veterinární ošetření
Krmivo	je skladováno na	lokaci
Krmivo	má	zásobu

3.2.3 Grafické znázornění základních vztahů

V rámci předchozí kapitoly byly identifikovány základní entity a jejich vzájemné vztahy. Pro lepší představu o budoucí databázi jsem si schematicky znázornil provázanost jednotlivých entit. Z následujícího obrázku vyplývá, které entity jsou nejvíce provázány a které jsou naopak spíše okrajové.



Obr. 14: Grafické znázornění vztahů mezi entitami

(Zdroj: vlastní zpracování)

3.2.4 Identifikace atributů základních entit

Následující tabulka obsahuje identifikaci základních atributů pro entity, které byly představeny v předchozích kapitolách.

Tab. 8: Identifikace atributů základních entit

(Zdroj: vlastní zpracování)

Entita	Jméno atributu	Popis
Chovatel	jméno chovatele	celé jméno chovatele
	email	email chovatele, který slouží jako login
	telefon	kontaktní telefon na chovatele
	heslo	přihlašovací heslo chovatele
	datum registrace	datum, kdy chovatel provedl registraci
	ovce	označení, jaká zvířata chce evidovat
	kozy	
	psi	
	drůbež	
Zvíře	registrační číslo	registrační číslo zvířete, pokud podléhá registraci
	číslo tetování	číslo tetování, pokud je zvíře tetováno
	číslo čipu	číslo čipu, pokud je zvíře očipované
	pohlaví	označení pohlaví zvířete
	jméno zvířete	volací jméno zvířete
	registrační jméno	registrační jméno zvířete
	datum narození	datum narození zvířete
	datum smrti	datum smrti zvířete
	porážková hmotnost	porážková hmotnost zvířete (pokud bylo poraženo)
	jméno chovatele	označení chovatele, od kterého bylo zvíře zakoupeno
	telefon chovatele	kontaktní telefon na původního chovatele
	email chovatele	kontaktní email na původního chovatele
	jméno majitele	jméno majitele, kterému bylo zvíře prodáno
	telefon majitele	kontaktní telefon na nového majitele
	email majitele	kontaktní email na nového majitele
Druh zvířete	název	název druhu zvířete

Plemeno zvířete	název	název plemene zvířete
Vlastnosti zvířete	název	název vlastnosti
	hodnota	hodnota vlastnosti
Lokace	název	název lokace
	použitelnost	označení, zda je možné lokaci používat
	karanténa	označení, zda je lokace v karanténě
Typ lokace	název	název typu lokace
Růstová tabulka	plemeno	označení, kterého plemene se týká
	pohlaví	označení, kterého pohlaví se týká
	věk	označení, věku zvířete, ke kterému se vztahuje
	minimální výška	minimální výška, které by zvíře v daném věku mělo dosahovat
	maximální výška	maximální výška, které by zvíře v daném věku mělo dosahovat
	minimální váha	minimální váha, které by zvíře v daném věku mělo dosahovat
	maximální váha	maximální váha, které by zvíře v daném věku mělo dosahovat
Aktivity zvířete	zvíře	označení zvířete, kterého se daná aktivita týká
	typ	typ dané aktivity
	datum a čas začátku	datum a čas začátku dané aktivity
	datum a čas konce	datum a čas konce dané aktivity
	místo	místo konání dané aktivity
Materiál	název	název materiálu
	měrná jednotka	měrná jednotka materiálu
Chovatelské akce	název	název chovatelské akce
	zvíře	označení zvířete, které se účastnilo chovatelské akce
	hodnocení	hodnocení, kterého zvíře dosáhlo
	hodnotitel	hodnotitel, který zvíře hodnotil
	typ	typ chovatelské akce
	místo	místo konání dané chovatelské akce
	datum a čas začátku	datum a čas začátku dané chovatelské akce
	datum a čas konce	datum a čas konce dané chovatelské akce
	organizátor	organizátor dané chovatelské akce
	telefon	kontaktní telefon

	email	kontaktní email
	odkaz na akci	odkaz na stránku dané chovatelské akce
Chovatelské stanice	název	název chovatelské stanice
	datum	datum registrace chovatelské stanice
	poradce	poradce chovu dané chovatelské stanice
	telefon poradce	kontaktní telefon na poradce chovatelské stanice
	email poradce	kontaktní email na poradce chovatelské stanice
Narození mláďat	zvíře	označení zvířete
	krycí list	pokud jsou mláďata narozena na základě krycího listu
	písmeno	písmenné označení vrhu narozených mláďat
	datum oplodnění	datum oplodnění samice
	očekávané datum	očekávané datum narození mláďat
	skutečné datum	skutečné datum narození mláďat
	datum potvrzení	datum potvrzení březosti
	očekávaný počet	očekávaný počet mláďat
	skutečný počet	skutečný počet narozených mláďat
Líhnutí drůbeže	druh vajec	označení druhu zvířete, jehož vejce jsou líhnuta
	datum začátku	datum začátku líhnutí
	očekávané datum konce	očekávané datum konce líhnutí
	skutečné datum konce	skutečné datum konce líhnutí
	množství vajec	množství líhnutých vajec
	datum první kontroly	datum první kontroly vajec
	vyřazený počet 1	počet vajec vyřazených při první kontrole
	datum druhé kontroly	datum druhé kontroly vajec
	vyřazený počet 2	počet vajec vyřazených při druhé kontrole
	množství	množství vylíhnutých jedinců
Krycí list	chovatelská stanice	označení chovatelské stanice
	zvíře	označení samice, pro kterou je krycí list vystaven
	datum žádosti	datum podání žádosti
	datum vystavení	datum vystavení krycího listu
	datum konce platnosti	datum konce platnosti krycího listu

	číslo listu	číslo krycího listu
	poradce chovu	poradce chovu
	kontaktní telefon	kontaktní telefon na poradce chovu
	kontaktní email	kontaktní email na poradce chovu
	plemeník 1 až 5	jména pěti vybraných plemeníků
	vybraný plemeník	plemeník, který byl vybrán k oplodnění
	datum oplodnění	datum uskutečnění pokusu o oplodnění
	březost	označení, zda je samice březí
Dojení zvířat	zvíře	označení zvířete, které bylo dojeno
	kvalita	kvalita nadojeného mléka
	datum a čas	datum a čas dojení
	množství	množství nadojeného mléka
Stříhání zvířat	zvíře	označení zvířete, které bylo stříháno
	kvalita	kvalita vlny, která byla získána stříháním
	datum	datum stříhání vlny
	množství	množství získané vlny
Veterinář	jméno	jméno veterináře
	telefon	kontaktní telefon na veterináře
	email	kontaktní email na veterináře
Veterinární ošetření	zvíře	označení zvířete, kterého se ošetření týká
	název	název ošetření
	lék	název léku využitého během ošetření
	cena	cena léčby
	datum začátku	datum začátku léčby
	datum konce	datum konce léčby
	datum karantény	datum konce karantény, pokud je vyžadována po provedení ošetření
	omezení mléka	označení, zda ošetření vede k omezení využití mléka
	omezení masa	označení, zda ošetření vede k omezení využití masa
	datum konce omezení	datum konce platnosti omezení využití masa a mléka
Krmivo	název	název krmiva
	popis	popis krmiva
	ovce	označení, pro které druhy zvířat lze krmivo používat
	kozy	

	psi	
	drůbež	
Zásoby	zásoba	označení konkrétní zásoby
	lokace	označení, kde se zásoba nachází
	množství	množství zásoby
	datum	datum vzniku zásoby
Finanční operace	typ	typ finanční operace
	položka	položka, které se záznam týká
	datum	datum uskutečnění finanční operace
	cena	cena, se kterou je položka spojena

3.3 Logický návrh databáze

Následující podkapitoly obsahují návrh struktury jednotlivých tabulek a ER diagram.

3.3.1 Návrh jednotlivých tabulek

Vzhledem k běžným konvencím v oblasti informatiky jsou jednotlivé tabulky nazvány anglicky. Struktura navrhovaných tabulek je popsána pomocí tabulek, které obsahují název atributu, jeho popis, datový typ, upřesnění velikosti použitého datového typu (sloupec Vel.), zda se jedná o primární (sloupec PK), či cizí klíč (sloupec FK) a zda může daný atribut nabývat hodnoty NULL. U primárních klíčů je uvedeno, zda jsou jejich hodnoty automaticky generovány (sloupec AI).

3.3.1.1 Tabulka 01_Users

Tato tabulka odpovídá identifikované entitě chovatel a slouží k ukládání dat spojených s uživateli. Jedná se konkrétně o: jméno uživatele, email, který zároveň bude sloužit jako login uživatele do webového rozhraní, heslo rovněž sloužící pro přihlášení do webového prostředí, datum registrace a jaké moduly bude uživatel v rámci řešení využívat.

Jedná se o základní tabulku navrhovaného řešení, neboť každý uživatel, který toto řešení bude chtít využívat bude muset nejprve provést registraci. Na tuto tabulku jsou následně navázány hodnoty všech parametrizovatelných číselníků a všechna evidovaná data o chovaných jedincích.

Tab. 9: Struktura tabulky 01 Users

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_User	jednoznačné označení uživatele	Integer		ano		ano	ne
Name	jméno uživatele	Varchar	100				ne
Email	email uživatele	Varchar	255				ne
Telephone	telefon uživatele	Varchar	15				ano
Password	heslo uživatele	Varchar	255				ne
Registration_ DateTime	datum a čas registrace uživatele	DateTime					ne
MDL_Sheep	zapnutí modulu ovce	Boolean					ne
MDL_Goat	zapnutí modulu koza	Boolean					ne
MDL_Dog	zapnutí modulu pes	Boolean					ne
MDL_Poultry	zapnutí modulu drůbež	Boolean					ne

3.3.1.2 Tabulka 02 Locations

Tato tabulka slouží k evidenci jednotlivých lokací, na kterých jsou zvířata chována a odpovídá identifikované entitě umístění. Z pohledu drobných chovatelů se jedná o přehled zemědělských budov, ohrad, výběhů aj.

Ke každé lokaci je evidováno jméno lokace, typ lokace, zda je aktuálně možné lokaci využívat a zda není aktuálně určena pro separaci zvířat, která nemohou být z důvodu karantény chována společně s ostatními.

Každý uživatel má možnost vytvořit si dle vlastních požadavků na členění lokací vlastní typy lokací a ty následně použít k vhodné segmentaci systému lokací.

Evidenci zvířat, krmiv aktuálně dostupných pro použití, materiálů, které má chovatel momentálně k dispozici, či produktů získaných chovatelskou činností, je vždy nutné provést na určitou lokaci, která reprezentuje konkrétní umístění dané evidované položky v databázi.

Tab. 10: Struktura tabulky 02 Locations

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Location	jednoznačné označení lokace	Integer		ano		ano	ne
ID_User	uživatel, kterému patří konkrétní lokace	Integer			ano		ne
ID_Location_Type	typ lokace	Integer			ano		ne
Location_Name	jméno lokace	Varchar	100				ne
Usable	označení, zda je možné lokaci používat	Boolean					ne
Quarantine	označení, zda je lokace v karanténě	Boolean					ne

3.3.1.3 Tabulka 03 Location_Types

Vzhledem k tomu, že jednotliví uživatelé mohou mít různé druhy lokací, mohou rovněž chtít tyto druhy rozlišovat. Z tohoto důvodu vznikl tento číselník, který může obsahovat označení typů lokací jako mohou být například chlév, pastvina aj, v případě, že uživatel má více umístění stejného druhu.

Každý uživatel si může vytvořit vlastní typ lokace dle vlastních potřeb, jelikož jsou podmínky u jednotlivých chovatelů různé.

V této tabulce bude evidován název druhu lokace, jeho identifikátor a uživatel, který tento typ definoval.

Tab. 11: Struktura tabulky 03 Location Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Location_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	uživatel, kterému náleží konkrétní typ	Integer			ano		ne
Type_Name	název lokace	Varchar	50				ne

3.3.1.4 Tabulka 04 Vets

Vzhledem k tomu, že s chovem zvířat je spojena i veterinární péče, je třeba v rámci navrhovaného řešení umožnit s ní související evidenci. Tato tabulka slouží k evidenci kontaktních údajů veterinárních lékařů, kteří tuto veterinární péči poskytují.

Každému chovateli poskytuje veterinární péči zpravidla více veterinářů, proto je v této tabulce nutné evidovat i konkrétní kontaktní údaje na daného veterinárního lékaře.

Tab. 12: Struktura tabulky 04 Vets
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Vet	jednoznačné označení veterináře	Integer		ano		ano	ne
ID_User	přiřazení veterináře ke konkrétnímu uživateli	Integer			ano		ne
Name	jméno veterináře	Varchar	100				ne
Telephone	telefonní číslo veterináře	Varchar	15				ano
Email	email veterináře	Varchar	255				ano

3.3.1.5 Tabulka 05 Vet_Actions

Jak bylo již dříve uvedeno, jednotlivé úkony mohou být prováděny různými veterinárními lékaři za různých podmínek.

Tato tabulka slouží k evidenci provedených ošetření, která byla jednotlivým zvířatům chovatele provedena. Některá ošetření mohou být spojena s nutností umístění zvířete do karantény a v případě hospodářských zvířat mohou být spojena s omezením využití živočišných produktů (maso a mléko).

Většinu ošetření je třeba veterinárním lékařům hradit, a proto je pro komplexní vyhodnocení nákladovosti chovaných zvířat třeba evidovat i cenu těchto ošetření. Cena prováděných úkonů je v čase variabilní a je závislá na druhu zvířete, kterému je daný úkon prováděn.

Tab. 13: Struktura tabulky 05 Vet Actions

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Vet_Action	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Vet	označení veterináře, který provedl daný úkon	Integer			ano		ne
Action_Name	název úkonu	Varchar	100				ne
Treatment_Name	název léku	Varchar	255				ano
Price	cena léčby	Decimal					ano
Start_Date	začátek léčby	Date					ano
End_Date	konec léčby	Date					ano
Quarantine_Date	konec karantény	Date					ano
Suspension_Date	konec platnosti omezení	Date					ano
Dairy_Suspension	omezení využití mléka	Boolean					ne
Meat_Suspension	omezení využití masa	Boolean					ne

3.3.1.6 Tabulka 06 Units_Of_Measure

Vzhledem k tomu, že v rámci tohoto řešení jsou evidovány různé entity, je třeba brát v potaz, že entity mohou mít různé měrné jednotky, a zároveň různí uživatelé mohou pro stejnou měrnou jednotku používat jiné označení. Například kusy (ks/pcs) a kilogramy (kg). Proto tato tabulka umožňuje definovat používané měrné jednotky.

Tab. 14: Struktura tabulky 06 Units Of Measure

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_UOM	jednoznačná identifikace záznamu	Integer		ano		ano	ne
ID_User	uživatel, který používá danou měrnou jednotku	Integer			ano		ne
UOM_Name	název měrné jednotky	Varchar	10				ne

3.3.1.7 Tabulka 07 Materials

S chovem zvířat je spojena potřeba různých chovatelských pomůcek, které lze označit jako materiál. Tento materiál může být různé povahy, některý materiál je průběžně spotřebováván (hračky pro psy), v jiných případech má jeho využívání dlouhodobější charakter (elektrické ohradníky).

Tato tabulka umožňuje evidovat různé druhy materiálů a jejich měrné jednotky, ve kterých uživatel tento materiál eviduje.

Tab. 15: Struktura tabulky 07 Materials
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Material	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele	Integer			ano		ne
ID_Material_UOM	měrná jednotka materiálu	Integer			ano		ne
Material_Name	název materiálu	Varchar	100				ne
Material_Description	popis materiálnu	Varchar	255				ano

3.3.1.8 Tabulka 08 Expositions

Vzhledem k tomu, že jednotlivá chovaná zvířata je možné i vystavovat v rámci různých chovatelských akcí, kterými mohou být výstavy, či veletrhy, slouží tato tabulka k zaznamenání základních propozic těchto akcí.

Mezi základní propozice výstavy patří místo a termín konání, její organizátor a informace potřebné k rozlišení druhu výstavy.

Pro jednotlivé chovatele je důležité mít možnost zaznamenat si webovou adresu výstavy, které se plánují zúčastnit, jelikož aktuality spojené s danou výstavou jsou zpravidla zveřejňovány pouze na stránkách pořadatele, v čase se mohou měnit a ne všechny informace jsou k dispozici v momentě přihlášení se na danou akci.

Tab. 16: Struktura tabulky 08 Expositions

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Exposition	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	přiřazení záznamu ke konkrétnímu uživateli	Integer			ano		ne
ID_Exposition_Type	druh výstavy	Integer			ano		ne
Exposition_Name	jméno výstavy	Varchar	100				ne
State	stát, kde se výstava koná	Varchar	100				ano
City	město, kde se výstava koná	Varchar	100				ano
Street_No	ulice, kde se výstava koná a číslo domu	Varchar	70				ano
Post_Code	poštovní směrovací číslo	Varchar	10				ano
Start_Date	den začátku výstavy	Date					ano
End_Date	den konce výstavy	Date					ano
Organizer	pořadatel výstavy	Varchar	255				ano
Telephone	telefon na pořadatele	Varchar	15				ano
Email	e-mail na pořadatele	Varchar	255				ano
Link	webový odkaz	Varchar	255				ano

3.3.1.9 Tabulka 09 Exposition_Types

Jednotlivé výstavy se mohou lišit, například rozsahem, či udělovanými tituly, a z tohoto důvodu je u každé výstavy třeba evidovat její typ.

Každý pořadatel výstavy má dle typu výstavy předem určeno, kolik výstav daného typu může v určitém období uspořádat (například předpisy dané chovatelské organizace, například klubu, stanovují maximální počet výstav za rok).

Tab. 17: Struktura tabulky 09 Exposition Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Exposition_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení, kterému uživateli daný typ náleží	Integer			ano		ne
Description	popis typu výstavy	Varchar	100				ne
International	rozdělení účastníků	Boolean					ne
National		Boolean					ne
Club		Boolean					ne

3.3.1.10 Tabulka 1001 Location_Feeding_Plans

Ze základních požadavků drobných chovatelů vyplynulo, že chtějí evidovat provedené krmení a zároveň predikovat, jaké množství krmiva budou v nejbližším období potřebovat, proto navrhované řešení umožňuje evidovat jednotlivé krmné dávky a jejich plán.

Plán je vždy vztažen ke konkrétní lokaci. Takovéto řešení zabezpečuje jak krmení zvířat chovaných a krmených samostatně (zejména psi), tak i krmení zvířat chovaných skupinově (ovce, kozy, slepice).

Krmení je pomocí uzpůsobení velkosti krmných dávek v související tabulce Condition Specific Feed možné vztahovat i k aktuálním podmínkám vnějšího okolí, či stavu zvířete.

V rámci uživatelského rozhraní bude následně zajištěna funkcionalita, která umožní automatické generování tohoto plánu pro určité období, v rámci, kterého uživatel bude moci provést výpočet pro konkrétní zvíře, případně konkrétní druh zvířat, či bude mít uživatel možnost omezit jednotlivé podmínky krmení, na základě, kterých má být výpočet proveden.

Tab. 18: Struktura tabulky 1001 Location Feeding Plans

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Location_Feeding_Plan	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Condition_Specific_Feed	zaznamenání stavu zvířat v lokaci	Integer			ano		ne
ID_User	označení, kterému uživateli plán náleží	Integer			ano		ne
ID_Location	označení lokace, ke které krmný plán náleží	Integer			ano		ne
ID_Feed	označení krmiva	Integer			ano		ne
Expected_Feeding_Date	datum krmení	Date					ne
Expected_Feeding_Time	čas krmení	Time					ne
Planned_Daily_Allowance	plánovaná spotřeba krmiva	Decimal					ne

3.3.1.11 Tabulka 1002 Location_Feeding_Entries

Jak již bylo výše uvedeno, je požadavkem chovatelů nejen plánovat krmné dávky, ale rovněž provádět evidenci již provedených krmení. Tato tabulka tedy slouží k zaevidování provedeného krmení.

Krmení jako takové může vycházet z plánovaného krmení, nicméně je možné evidovat i neplánované krmení (například mimořádné). Uživatel ke každému krmení zaznamená, kolik krmiva spotřeboval a přesné datum a čas, kdy bylo krmení provedeno. Je tedy možné zaevidovat i několik krmení v rámci jednoho dne.

Tab. 19: Struktura tabulky 1002 Location Feeding Entries

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Feeding_Entry	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Feeding_Plan	označení, kterému krmnému plánu záznam náleží	Integer			ano		ano
Feeded_Amount	zkrmené množství	Decimal					ne
Feeding_Date	den, kdy bylo krmení uskutečněno	Date					ne
Feeding_Time	čas, kdy bylo krmení uskutečněno	Time					ne

3.3.1.12 Tabulka 1003 Condition_Specific_Feed

Každému zvířeti mohou být v průběhu dne podávána různá krmiva v různě velkých krmných dávkách (lišit se může například dávka granulovaného krmiva a dávka podávaných doplňkových látek).

Stejně tak může každé zvíře vyžadovat speciální dávku krmení na základě jeho aktuálního stavu (například zdravotní omezení), či ve vztahu k jeho váze (při nadváze jedince zpravidla u některých druhů zvířat dochází k redukci krmné dávky, v jejímž důsledku se očekává snížení hmotnosti jedince).

V případě stádových zvířat (například ovce), kdy není krmení vztaženo ke konkrétnímu jedinci, ale k jejich skupině, bude v této tabulce zadané množství, které má být zkrmeno na jednoho jedince daného druhu při splnění dané podmínky. Funkcionalita v rámci uživatelského rozhraní následně zajistí, že v rámci generování plánu krmení pro určitou lokaci, bude dle aktuálního počtu zvířat ve zvolené lokaci vypočtena dávka pokrývající krmení všech těchto jedinců.

Tab. 20: Struktura tabulky 1003 Condition Specific Feed

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Condition_Specific_Feed	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Feed	označení krmiva pro zvíře	Integer			ano	ano	ne
ID_Feeding_Conditions	podmínky krmení zvířete	Integer			ano		ne
ID_Animal_Type	druh krmeného zvířete	Integer			ano		ne
ID_Animal	označení konkrétního zvířete	Integer			ano		ano
Minimal_Weight	minimální váha	Decimal					ne
Maximal_Weight	maximální váha zvířete	Decimal					ne
Daily_Allowance	krmná dávka na den	Decimal					ne

3.3.1.13 Tabulka 1004 Feed

Krmivem v pojetí navrhovaného řešení nejsou pouze samotná hlavní krmiva (granulovaná krmiva, jádro, seno, či další), ale i případné doplňkové složky krmiva, které mohou být chovaným zvířatům podávány (například vitamíny, či minerály).

Každé krmivo může být využito pro krmení jednoho, či více druhů zvířat, což je typické zejména pro hospodářská zvířata.

Proto je pro každé krmivo nutné specifikovat, pro jaké moduly, reprezentující druhy chovaných zvířat, může být použito. Neboť u krmiv pro hospodářská zvířata je krmivo typicky možné využít pro více druhů zvířat.

Stejně tak řešení umožňuje evidovat krmiva v různých měrných jednotkách, jelikož krmiva se mohou lišit nejen velikostí balení (v rámci řešení je reprezentováno měrnou jednotkou), tak i strukturou (kapalné, či pevné).

Tab. 21: Struktura tabulky 1004 Feed

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Feed	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení, ke kterému uživateli krmivo náleží	Integer			ano		ne
ID_Feed_UOM	měrná jednotka krmiva	Integer			ano		ne
Feed_Name	název krmiva	Varchar	255				ne
Feed_Description	popis krmiva	Varchar	255				ano
MDL_Sheep	označení, pro jaké zvíře může být krmivo použito	Boolean					ne
MDL_Goat		Boolean					ne
MDL_Dog		Boolean					ne
MDL_Poultry		Boolean					ne

3.3.1.14 Tabulka 1005 Feeding_Conditions

Vzhledem k tomu, že každý chovatel může pro uzpůsobení krmných dávek používat jiný mechanismus, umožňuje řešení definovat uživatelské podmínky uzpůsobení krmných dávek, včetně definice koeficientu, který je následně při výpočtu krmných dávek využit.

Tab. 22: Struktura tabulky 1005 Feeding Conditions

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Feeding_Condition	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému dané podmínky náleží	Integer			ano		ne
Feeding_Condition_Description	popis podmínečného krmení	Varchar	255				ne
Coeficient	koeficient pro výpočet krmné dávky	Decimal					ne

3.3.1.15 Tabulka 2001 Location_Inventory_Entries

Tato tabulka slouží k souhrnné evidenci zásob v čase. Hlavními evidovanými druhy zásob jsou: krmiva, materiál využívaný k zajištění provozu a produkty získané chovem zvířat.

Data do této tabulky nebudou vkládána chovatelem přímo, ale budou vznikat automatizovaně při samotné práci s uživatelským rozhraním. Uživatel tedy pouze uvede typ pohybu, který chce provést a systém na jeho základě provede korekci zásob.

Tab. 23: Struktura tabulky 2001 Location Inventory Entries
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Location_Inventory_Entry	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému záznam náleží	Integer			ano		ne
ID_Location	označení lokace	Integer			ano		ne
ID_Entry_Type	typ záznamu	Integer			ano		ne
ID_Source_Type	označení původu položky	Integer			ano		ano
ID_Source_UOM	měrná jednotka, ve které byla položka zaevidována	Integer			ano		ano
Number_Source_UOM	počet měrných jednotek	Decimal					ne
Remaining_Quantity	zůstatkové množství položek na skladě	Decimal					ne
Entry_Date	datum vzniku položky	Date					ne
Usable	označení, zda je položka použitelná	Boolean					ne
Open	označení, zda je položka na skladě	Boolean					ne
Source_Entry_No	označení položky v tabulce	Integer					ano

3.3.1.16 Tabulka 2002 Source_Types

Tabulka Source Types je systémovou tabulkou, které je společná pro všechny uživatele. Jelikož v tabulkách Location Inventory Entries a Financial Entries dochází ke slučování údajů z více zdrojů (tabulek), slouží tato tabulka ke specifikaci zdroje, ze kterého konkrétní údaj pochází.

Hlavním účelem existence této tabulky jsou statistická vyhodnocování jednotlivých typů pohybů.

Tab. 24: Struktura tabulky 2002 Source Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Source_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
Table	název tabulky	Varchar	100				ne

3.3.1.17 Tabulka 2003 Entry_Types

Tabulka Entry Types je stejně jako předchozí tabulka Source Types systémovou tabulkou. Tato tabulka slouží k definování typu pohybu, kterým k vzniku položky v tabulkách Location Inventory Entries a Financial Entries došlo.

K základním typům pohybů, které bude v rámci řešení možné evidovat, patří příjem, výdej, či přesun.

Tab. 25: Struktura tabulky 2003 Entry Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Entry_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
Entry_Type_Description	popis záznamu	Varchar	100				ne

3.3.1.18 Tabulka 2004 Financial_Entries

Jedním z požadavků drobných chovatelů bylo získat přehled o finanční tocích spojených s jejich chovatelskou činností. Tato tabulka slouží k evidenci všech finančních nákladů a výnosů spojených s chovem zvířat.

Náklady typicky vznikají při pořízení zvířat, péči o ně, při pořízení chovatelských pomůcek (materiálu), či při nákupu potřebných krmiv. Výnosy spojené s chovem mohou vznikat například prodejem zvířat, či produktů získaných chovem.

Tab. 26: Struktura tabulky 2004 Financial Entries

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Financial_Entry	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele	Integer			ano		ne
ID_Location_Inventory_Entry	označení lokace	Integer			ano		ne
ID_Entry_Type	typ položky	Integer			ano		ne
ID_Source_Type	zdroj záznamu	Integer			ano		ano
ID_Cost_Type	členění nákladů	Integer			ano		ano
ID_Sales_Type	členění příjmů	Integer			ano		ano
Source_Entry_No	Položka	Integer					ano
Entry_Date	datum vzniku záznamu	Date					ne
Cost_Amount	výše nákladu	Decimal					ano
Sales_Amount	výše příjmu	Decimal					ano

3.3.1.19 Tabulka 2005 Cost_Types

Jak již bylo výše uvedeno, chov je spojen s velkým množstvím nákladů různého druhu. Každý chovatel může mít odlišné požadavky na vyhodnocování nákladů spojených s chovem, například dle typu chovaného zvířete, či způsobu vzniku nákladu. K podchycení všech těchto variant slouží tato tabulka, která je uživatelsky parametrizovatelným číselníkem.

Tab. 27: Struktura tabulky 2005 Cost Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Cost_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému záznam náleží	Integer			ano		ne
Cost_Type_Description	Popis	Varchar	255				ne

3.3.1.20 Tabulka 2006 Sales_Types

Kromě nákladů je chov zvířat spojený i s výnosy. Obdobně jako u nákladů je tedy potřeba umožnit chovatelům evidovat dosažené výnosy v požadované struktuře a na jejich základě vyhodnocovat rentabilitu chovu.

Za tímto účelem byla do řešení zahrnuta tato tabulka, která je obdobně jako tabulka Cost Types uživatelsky parametrizovatelným číselníkem.

Tab. 28: Struktura tabulky 2006 Sales Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Sales_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému záznam náleží	Integer			ano		ne
Sales_Type_Description	Popis	Varchar	255				ne

3.3.1.21 Tabulka 10001 Animals

Tato tabulka je hned po tabulce Users druhá nejdůležitější. Každý chovatel může chovat libovolné množství zvířat.

V rámci návrhu databáze pracuji s tím, že chovatelé, kteří projevili o řešení zájem, chovají ovce, kozy, psy a drůbež, proto byla struktura této tabulky uzpůsobena tak, aby umožňovala zaznamenat data, která potřebují chovatelé těchto skupin zvířat evidovat. Nicméně řešení bude dle potřeby následně možné rozšířit o další moduly.

Drobné rozdíly dat, evidovaných o jednotlivých skupinách zvířat, budou řešeny na úrovni uživatelského rozhraní na základě modulu, se kterým uživatel právě pracuje. Jako příklad této skutečnosti lze uvést registrační jméno zvířete, které se vztahuje pouze k modulu psů, případně porážková hmotnost zvířete, která se vztahuje pouze k hospodářským zvířatům (psi typicky poráženi nejsou).

V rámci evidence zvířat je třeba evidovat, o jaký druh a plemeno zvířete se jedná. Pokud má evidovaný jedinec registrační číslo, čip, či tetování, pak je nutné evidovat tato identifikační čísla.

Kromě jiného je důležité stanovit typ chovnosti zvířete. Zvíře může být vhodné, či nevhodné k chovu, případně ještě chovné s omezením. Na základě typu chovnosti může být omezen počet odchovů mláďat, které může zvíře mít (typické pro psy), či u hospodářských zvířat může být zvíře určeno k porážce.

Dále je v rámci navrhovaného řešení možné evidovat, zda je zvíře naživu, případně jaká byla příčina úmrtí. Tyto informace jsou důležité k vyhodnocování úmrtnosti jedinců dle zvoleného typu úmrtí.

Chovatel může evidovat i zvířata, jejichž vlastníkem není, v takovém případě je možné zaevidovat majitele zvířete. V případě, že zvíře zakoupil, může pak evidovat informace o chovateli, který mu zvíře prodal.

Na tuto tabulku je navázáno velké množství číselníků, jejichž hodnoty jsou v některých případech naplňovány při zaevidování zvířete (plemeno a druh) a v některých případech jsou doplňovány až v průběhu života jedince (typicky chovnost, či typ úmrtí). Řešení pomocí číselníků bylo zvoleno kvůli přehlednosti řešení, možnosti uživatelsky definovat požadované hodnoty a možnostem filtrováním pro následné vyhodnocování zaevidovaných dat.

Tab. 29: Struktura tabulky 10001 Animals

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	přiřazení záznamu ke konkrétnímu uživateli	Integer			ano		ne
ID_Animal_Type	označení typu zvířete	Integer			ano		ne
ID_Death_Type	typ smrti zvířete	Integer			ano		ano
ID_Breeding_Type	omezení chovnosti	Integer			ano		ne
ID_Breed_Type	označení plemene	Integer			ano		ne
ID_UOM	měrná jednotka	Integer			ano		ne
Registration_Number	registrační číslo zvířete	Varchar	50				ano
Calling_Name	jméno zvířete	Varchar	50				ano
Registration_Name	registrační jméno zvířete	Varchar	150				ano
Sex	pohlaví zvířete	Boolean					ano
Birth_Date	datum narození zvířete	Date					ano
Death_Date	datum smrti zvířete	Date					ano
Death_Weight	porážková váha zvířete	Decimal					ano
Alive	označení, zda je zvíře živé	Boolean					ano
Owned	označení, zda zvíře patří chovateli	Boolean					ano
Tatto_Number	číslo tetování	Varchar	60				ano
Chip_Number	číslo čipu	Varchar	60				ano
Breeder_Name	jméno chovatele	Varchar	100				ano
Breeder_Telephone	telefon chovatele	Varchar	15				ano
Breeder_Email	email chovatele	Varchar	255				ano
Owner_Name	jméno majitele	Varchar	100				ano
Owner_Telephone	telefon majitele	Varchar	15				ano
Owner_Email	email majitele	Varchar	255				ano

3.3.1.22 Tabulka 10002 Animal_Types

Vzhledem k tomu, že v současné době je v rámci řešení uvažována realizace čtyř modulů, které zahrnují větší množství druhů zvířat, je třeba tyto druhy rozlišovat. Zatímco některým modulům náleží pouze jeden druh (kozy), jiné moduly zahrnují více druhů zvířat (například drůbež, do které mohou spadat například slepice, křepelky, či kachny).

V důsledku toho je třeba, aby uživatel definoval druhy zvířat, které chová, a specifikoval, pod jaký modul daný druh spadá. Tento číselník je stěžejní pro rozlišování mezi moduly na úrovni uživatelského rozhraní.

Tab. 30: Struktura tabulky 10002 Animal Types
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Animal_Type	název druhu zvířete	Varchar	50				ne
MDL_Dog	označení, pod který modul spadá konkrétní druh	Boolean					ne
MDL_Sheep		Boolean					ne
MDL_Goat		Boolean					ne
MDL_Poultry		Boolean					ne

3.3.1.23 Tabulka 10003 Death_Types

Jelikož jedním ze základních požadavků chovatelů bylo vyhodnocovat úmrtnost zvířat, je třeba evidovat, že dané zvíře zemřelo, což je řešeno v tabulce Animals, a zároveň evidovat příčinu jeho úmrtí.

Jednotlivá zvířata mohou umírat různými způsoby, některé způsoby mohou být společné pro všechny skupiny zvířat (stáří, nemoc), zatímco jiné jsou specifické pouze pro některé skupiny (porážka). Podrobnost evidence se může lišit pro jednotlivé uživatele.

Tab. 31: Struktura tabulky 10003 Death Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Death_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Death_Type	název typu smrti	Varchar	50				ne
MDL_Dog	označení, pod který modul spadá konkrétní záznam	Boolean					ne
MDL_Sheep		Boolean					ne
MDL_Goat		Boolean					ne
MDL_Poultry		Boolean					ne

3.3.1.24 Tabulka 10004 Breeding_Types

V zájmovém chovu se mohou vyskytovat jak zvířata vhodná pro chov, tak i zvířata nevhodná pro chov. U některých skupin zvířat může být chovnost podmíněna splněním určitých podmínek, například u psů věk feny, či splnění podmínek uchovnění definovaných daným chovatelským klubem. U hospodářských zvířat zpravidla o vhodnosti k chovu rozhoduje chovatel sám, takovéto rozhodnutí může být například motivováno snahou eliminovat některé nežádoucí vlastnosti.

Tab. 32: Struktura tabulky 10004 Breeding Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Breeding_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
ID_Animal_Type	označení, kterému druhu daný záznam náleží	Integer			ano		ne
Breeding_Type	název typu chovnosti	Varchar	50				ne

3.3.1.25 Tabulka 10005 Tabulka Breed_Types

V rámci jednotlivých chovaných druhů zvířat může chovatel chovat různé množství plemen.

Z tohoto důvodu je třeba chovateli umožnit zaevidovat všechna plemena, která chová, a zařadit je pod příslušný druh zvířete, čímž bude v rámci uživatelského rozhraní následně umožněn jejich výběr při práci s daným modulem.

Tab. 33: Struktura tabulky 10005 Breed Types
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Breed_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal_Type	označení druhu zvířete	Integer			ano		ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Breed_Type	název plemene zvířete	Varchar	150				ne

3.3.1.26 Tabulka 10006 Animal_Features

Různá chovaná zvířata v závislosti na skupině, do které patří, mohou mít různé vlastnosti, které chtějí chovatelé evidovat. Tyto vlastnosti mohou sloužit pro výběr vhodných jedinců k dalšímu chovu.

Tato tabulka slouží k zaznamenání vlastností konkrétního zvířete. Může se jednat například o exteriérové vlastnosti zvířete, jako je například barva očí, či barva srsti, povahové vlastnosti zvířete, případně jakékoliv jiné další vlastnosti zvířete, které chce chovatel evidovat (například alergie).

Tato tabulka tedy slouží k provázání jedince s vlastností a její konkrétní hodnotou. Příkladem může být pes, u kterého chce chovatel evidovat barvu srsti, která může být buď černá, nebo plavá, potom by záznam v této tabulce říkal, že konkrétní pes má srst v černé barvě.

Tab. 34: Struktura tabulky 10006 Animal Features

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Feature	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Feature	označení vlastnosti	Integer			ano		ne
ID_Feature_Value	hodnota vlastnosti	Integer			ano		ne
Start_Date	datum začátku	Date					ne
End_Date	datum konce	Date					ano

3.3.1.27 Tabulka 10007 Features

Každý chovatel má možnost si dle svých vlastních potřeb definovat vlastnosti, které chce u svých zvířat evidovat. Tyto vlastnosti jsou vztažené ke konkrétnímu druhu zvířat (u psů chce chovatel evidovat jiné vlastnosti než u ovcí). Typickou vlastností může být například u psů barva srsti a u ovcí jejich reprodukční vlastnosti.

Tab. 35: Struktura tabulky 10007 Features

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Feature	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal_Type	druh zvířete	Integer			ano		ne
Feature_Name	název vlastnosti	Varchar	100				ne
Feature_Description	popis vlastnosti	Varchar	255				ano

3.3.1.28 Tabulka 10008 Feature_Values

Každá vlastnost definovaná chovatelem může nabývat určité hodnoty z konečné množiny hodnot. Tuto množinu hodnot musí uživatel specifikovat v této tabulce. Příkladem hodnot, kterých může nabývat vlastnost barva srsti, může být například černá, hnědá či plavá.

Tab. 36: Struktura tabulky 10008 Feature Values

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Feature_Value	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Feature	označení vlastnosti	Integer			ano		ne
Feature_Value_Description	popis hodnoty vlastnosti	Varchar	100				ne

3.3.1.29 Tabulka 10009 Animal_Locations

Jedním z požadavků uživatelů je mít přehled o tom, kde se vybraný jedinec v současné době nachází. Vzhledem k tomu, že se jednotlivá zvířata mohou v čase přemisťovat, je třeba kromě prvotního umístění jedince evidovat i změny jeho umístění v čase.

Tab. 37: Struktura tabulky 10009 Animal Locations

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Location	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Location	označení lokace	Integer			ano		ne
Start_Date	datum začátku	Date					ne
End_Date	datum konce	Date					ano
Active	označení platnosti záznamu	Boolean					ne
Note	poznámka	Varchar	255				ano

3.3.1.30 Tabulka 10010 Daily_Dairy

U některých evidovaných druhů zvířat (ovce, kozy) může chovatel provádět dojení. V rámci dojení je třeba evidovat, jaké zvíře bylo dojeno, kdy samotné dojení proběhlo, jaké množství mléka a v jaké kvalitě bylo takto získáno.

Tab. 38: Struktura tabulky 10010 Daily Dairy

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Daily_Dairy	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Dairy_Quality	kvalita mléka	Integer			ano		ne
ID_UOM	měrná jednotka	Integer			ano		ne
Dairy_DateTime	den a čas dojení	Datetime					ne
Milk_Quantity	množství nadojeného mléka	Decimal					ne

3.3.1.31 Tabulka 10011 Dairy_Quality

V rámci jednotlivých dojení je třeba rozlišovat, jakou kvalitu má nadojené mléko. Kvalita mléka může být ovlivněna například zdravotním stavem jedince. Každý chovatel může mít jinou úroveň rozlišení kvality mléka, proto má chovatel možnost nastavit si pomocí této tabulky jednotlivé kvality.

Hlavním účelem rozlišení jednotlivých kvalit je možnost statistického vyhodnocování nadojeného mléka.

Tab. 39: Struktura tabulky 10011 Dairy Quality

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Dairy_Quality	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Dairy_Quality	označení kvality mléka	Varchar	50				ne

3.3.1.32 Tabulka 10012 Wool_Cutting

Zvířata, která jsou evidována v rámci některých navržených modulů (ovce), vyžadují pravidelné provádění stříhání vlny. Ke konkrétnímu stříhání má uživatel možnost zaevidovat, jaké množství vlny a v jaké kvalitě bylo získáno. Pokud chovatel takto získanou vlnu dále nezpracovává, není nutné evidovat množství a kvalitu vlny.

Tab. 40: Struktura tabulky 10012 Wool Cutting

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Wool_Cutting	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Wool_Quality	kvalita vlny	Integer			ano		ano
ID_UOM	měrná jednotka	Integer			ano		ne
Date	datum stříhání	Date					ne
Wool_Quantity	množství vlny	Decimal					ano

3.3.1.33 Tabulka 10013 Wool_Quality

Pokud chce chovatel provádět další zpracování vlny, může být žádoucí sledovat její kvalitu. Každý chovatel může rozlišovat různé kvality vlny.

Tab. 41: Struktura tabulky 10015 Wool Quanlity

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Wool_Quality	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Wool_Quality	označení kvality vlny	Varchar	50				ne

3.3.1.34 Tabulka 10014 Kernels

U zvířat, u kterých chce chovatel odchovávat zvířata s průkazem původu (psi), je třeba si zažádat o mezinárodně chráněný název chovatelské stanice a pod touto stanicí provést následný odchov mláďat.

V případě chovu čistokrevných zvířat je často nutné konzultovat průběh procesu odchovu zvířat s poradcem chovu. Ve většině případů je obvyklé, že jeden chovatel má pouze jednu chovatelskou stanici.

Tab. 42: Struktura tabulky 10014 Kernels
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Kernel	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	přiřazení k uživateli	Integer			ano		ne
Name	jméno chovatelské stanice	Varchar	100				ne
Registration_Date	datum registrace	Date					ano
Advisor	poradce chovu	Varchar	100				ano
Advisor_Telephone	telefon poradce chovu	Varchar	15				ano
Advisor_Email	email poradce chovu	Varchar	255				ano

3.3.1.35 Tabulka 10015 Animal_Kernels

Pokud chce chovatel chovat psy s průkazem původu, jak již bylo uvedeno výše, musí si zřídit chovatelskou stanici.

Odchov štěňat většinou probíhá pod chovatelskou stanicí majitele feny. Z tohoto důvodu je tudíž potřeba provázat fenu s konkrétní chovatelskou stanicí, čehož je docíleno pomocí této tabulky.

Chovatelská stanice není vázána na konkrétní plemeno zvířete, proto není nijak omezeno, jaké jedince je možné s chovatelskou stanici provázat.

Tab. 43: Struktura tabulky 10015 Animal Kernels

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Kernel	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Kernel	označení chovatelské stanice	Integer			ano		ne
ID_Animal	označení zvířete	Integer			ano		ne
Start_Date	začátek platnosti záznamu	Date					ne
End_Date	konec platnosti záznamu	Date					ano

3.3.1.36 Tabulka 10016 Breeding_Permit

V rámci chovu čistokrevných zvířat je nutné zažádat o vystavení krycího listu. Vystavení krycího listu zpravidla zajišťuje příslušný klub.

Krycí list zamezuje spojení jedinců, které by mohlo vést k porušení pravidel chovu daného plemene.

Mezi základní faktory, které se jednotlivé kluby snaží eliminovat, patří příbuznost jedinců, či přenos dědičných onemocnění, jako jsou dysplazie kyčelních kloubů (DKK), či dysplazie loketních kloubů (DLK).

Na krycím listu jsou uvedeni plemeníci, kterými má majitel chovné feny v úmyslu danou fenu nakrýt. Plemeníci nejsou primárně stanovováni chovatelským klubem a majitel si je zajišťuje sám. Klub nicméně zodpovídá za dodržení podmínek stanovených výše, kdy vystavením krycího listu stvrzuje, že daní plemeníci jsou pro nakrytí uvedené feny vhodní.

Krycí list bývá zasílán chovateli v tištěné podobě v několika výtiscích, které v různém stádiu krytí, či následné březosti chovné feny, zasílá majitel feny zpět oblastnímu poradci chovu.

Krycí list má zpravidla omezenou platnost (například šest měsíců od vystavení), po této době jej již není možné použít.

Tab. 44: Struktura tabulky 10016 Breeding Permits

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Breeding_Permit	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal_Kernel	označení zvířete v chovatelské stanic	Integer			ano		ne
Request_Date	datum podání žádosti	Date					ne
Issue_Date	datum vystavení krycího listu	Date					ano
Expiration_Date	datum expirace krycího listu	Date					ano
Active	označení, zda lze krycí list použít	Boolean					ano
Permit_Number	číslo krycího listu	Varchar	20				ano
Breeding_Advisor	poradce chovu	Varchar	100				ano
Advisor_Telephone	kontaktní telefon	Varchar	15				ano
Advisor_Email	kontaktní email	Varchar	255				ano
Male 1	plemeník 1	Varchar	150				ano
Male 2	plemeník 2	Varchar	150				ano
Male 3	plemeník 3	Varchar	150				ano
Male 4	plemeník 4	Varchar	150				ano
Male 5	plemeník 5	Varchar	150				ano
Act_Date	datum krytí	Date					ano
Breeding_Male	vybraný plemeník	Varchar	150				ano
Fertility	březost	Boolean					ano

3.3.1.37 Tabulka 10017 Animal_Expositions

Vzhledem k tomu, že v rámci zájmového chovu se mohou chovaní jedinci účastnit výstavních akcí, je třeba umožnit evidenci chovatelských akcí, kterých se chovaný jedinec zúčastnil a jakého na nich dosáhl hodnocení.

Tab. 45: Struktura tabulky 10017 Animal Expositions

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Exposition	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Expositon	označení výstavy	Integer			ano		ne
Date	datum	Date					ano
Result	hodnocení	Varchar	150				ano
Referee	rozhodčí	Varchar	100				ano

3.3.1.38 Tabulka 10018 Animal_Examinations

U psů je typické, že v rámci absolvování výcvikových aktivit dochází ke skládání různých zkoušek prokazujících úroveň jejich schopností. Zkoušky mohou být různého typu a v závislosti na tom se vyžaduje jiná úroveň schopností. Ke každé absolvované zkoušce je třeba evidovat její místo, rozhodčího, který jedince posuzoval, a dosažené hodnocení.

Tab. 46: Struktura tabulky 10018 Animal Examinations

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Examination	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Examination_Type	typ zkoušky	Integer			ano		ne
ID_Animal	označení zvířete	Integer			ano		ne
Date	datum konání	Date					ano
City	město, kde se zkouška koná	Varchar	100				ano
Street_No	ulice, kde se zkouška koná a číslo domu	Varchar	70				ano
Result	splněno/nesplněno	Boolean					ano
Result_Code	kód výsledku	Varchar	100				ano
Handler	psovod	Varchar	100				ano
Referee	rozhodčí	Varchar	100				ano

3.3.1.39 Tabulka 10019 Examination_Types

Zkoušky, které mohou jedinci absolvovat, jsou různého typu. V některých případech je složení vyššího typu zkoušky podmíněno složením nižší náročnosti zkoušky.

Složení jednotlivých zkoušek má vliv na atraktivnost jedince v rámci chovu a jeho zařazení do třídy v rámci výstav, kterých se účastní.

Tato tabulka je uživatelsky parametrizovatelným číselníkem jednotlivých typů zkoušek, které mohou jedinci daného chovatele skládat.

Tab. 47: Struktura tabulky 10019 Examination Types
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Examination_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Name	název zkoušky	Varchar	100				ne
Description	popis zkoušky	Varchar	255				ano

3.3.1.40 Tabulka 10020 Growth_Table

U jednotlivých chovaných zvířat může být žádoucí sledovat jejich výšku a váhu v průběhu růstu jedince.

Tato tabulka slouží k zaznamenání optimálních hodnot v určitém věku. Na základě údajů v této tabulce může chovatel uzpůsobit krmnou dávku pro konkrétního jedince, a to například v případě, že hmotnost výrazně přesahuje obvyklou hodnotu.

Tyto údaje mohou být velmi užitečné například pro začínající chovatele, kteří nemají s daným plemenem předchozí zkušenosti, a proto je pro ně obtížné vývoj jedince vhodně korigovat.

Tab. 48: Struktura tabulky 10020 Growth Table

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Growth_Record	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Breed_Type	označení plemene	Integer			ano		ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Sex	pohlaví	Boolean					ne
Age	věk	Varchar	20				ne
Weight_min	minimální váha	Decimal					ano
Weight_max	maximální váha	Decimal					ano
Height_min	minimální výška	Decimal					ano
Height_max	maximální výška	Decimal					ano

3.3.1.41 Tabulka 10021 Animal_Meassurements

Pro jednotlivé jedince může být požadovaná evidence vývoje jejich růstu a přibytku hmotnosti v čase. Tato tabulka slouží k zaznamenání skutečné váhy a výšky jedince v určitou dobu.

Využití údajů zaznamenaných v této tabulce se může lišit dle skupiny, do které spadá chované zvíře. Zatímco u psů může sloužit k optimalizaci krmné dávky, u hospodářských zvířat může sloužit k výběru jedinců vhodných k porážce.

Tab. 49: Struktura tabulky 10023 Animal Meassurements

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Animal_Meassurement	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
Age	věk	Varchar	20				ano
Date	datum měření	Date					ne
Weight	váha zvířete při měření	Decimal					ano
Height	výška zvířete při měření	Decimal					ano

3.3.1.42 Tabulka 10022 Litter

Chovatelé, kteří si pro chov zřídili chovatelskou stanici, pro tuto chovatelskou stanici potřebují evidovat jednotlivé vrhy, které v jejím rámci realizovali. Vrh je vždy vázaný na konkrétní samici, nicméně pořadí vrhu je dáno chovatelskou stanicí.

U psů je typické, že jsou jednotlivé vrhy označovány písmeny abecedy, které označují pořadí daného vrhu. Průběh březosti feny je možné průběžně sledovat díky moderním vyšetřovacím metodám (ultrazvuk), které umožňují potvrzení březosti a odhad počtu narozených mláďat.

V této tabulce budou rovněž evidovány vrhy hospodářských zvířat, které nejsou označovány a chovatel pouze chce evidovat počty narozených mláďat.

Tab. 50: Struktura tabulky 10022 Litters

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Litter	jednoznačné označení chovatelské stanice	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Breeding_Permit	označení krycího listu	Integer			ano		ano
Letter	písmenné označení vrhu	Varchar	2				ano
Litter_Order	pořadí vrhu	Tinyint					ano
Inpregnation_Date	datum oplodnění	Date					ano
Actual_Birth_Date	datum porodu	Date					ano
Fertility_Confirm_Date	potvrzení březosti	Date					ano
Expected_Birth_Date	očekávané datum porodu	Date					ano
Expected_Animal_Amount	očekávaný počet mláďat	Tinyint					ano
Actual_Animal_Amount	skutečný počet mláďat	Tinyint					ano

3.3.1.43 Tabulka 10023 Litter_Animals

V rámci jednoho vrhu se může narodit několik jedinců, jejichž počet je zpravidla dán obvyklými hodnotami pro daný druh zvířete a jeho plemeno.

Tato tabulka slouží k propojení jednotlivých mláďat s vrhem, v rámci, kterého se narodila. Díky příslušnosti mláďat k vrhu je možné vyhodnocovat například úmrtnost jedinců při porodu.

V některých případech, zejména u psů, je požadováno označení jednotlivých narozených mláďat a zaznamenání tohoto označení. Může se jednat o barvu stužky, či obojku, kterou byl jedinec označen po narození.

Tab. 51: Struktura tabulky 10023 Litter Animals
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Litter_Animal	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal	označení zvířete	Integer			ano		ne
ID_Litter	označení vrhu	Integer			ano		ne
Birth_DateTime	čas narození mláděte	Datetime					ano
Mark	označení jedince (stužka, obojek, včetně barvy)	Varchar	50				ano

3.3.1.44 Tabulka 10024 Activity_Calendar

Kromě aktivit spojených se samotným chovem, jako jsou výstavy, zkoušky, či aktivity potřebné pro uchovnění (svod mladých, či bonitace), je například u psů žádoucí provádět i jiné aktivity. Může jít například o pravidelný výcvik psa.

K plánování těchto aktivit slouží následující tabulka. Jednotlivé aktivity mohou být různého typu a mohou se konat v různých termínech s různou pravidelností. Zatímco některých aktivit se může chovatel účastnit poměrně pravidelně, některé aktivity mohou být výjimečné.

Tab. 52: Struktura tabulky 10024 Activity Calendar

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Activity_Calendar	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	uživatel	Integer			ano		ne
ID_Animal	zvíře	Integer			ano		ne
ID_Activity_Type	označení typu aktivity	Integer			ano		ne
Activity_StartDate	začátek aktivity	Datetime					ano
Activity_EndDate	konec aktivity	Datetime					ano
City	město, kde se aktivita koná	Varchar	100				ano
Street_No	ulice, kde se aktivita koná a číslo domu	Varchar	70				ano
Post_Code	psč města, kde se aktivita koná	Varchar	10				ano
State	země, kde se aktivita koná	Varchar	100				ano

3.3.1.45 Tabulka 10025 Activity_Types

Tato tabulka je číselníkem typů možných aktivit. Může se jednat například o skupinový výcvik, výcvik jednotlivce, případně speciální aktivity typu hoopers, či agility.

Tab. 53: Struktura tabulky 10025 Activity Types

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Activity_Type	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele, kterému daný záznam náleží	Integer			ano		ne
Activity_Name	název aktivity	Varchar	255				ne
Activity_Cost	cena aktivity	Decimal					ano

3.3.1.46 Tabulka 10026 Eggs_Locations

Jelikož je drůbež chována ve skupině, není možné přesně identifikovat jedince, který snesl dané vejce, nicméně je možné identifikovat druh zvířete, který je jeho původcem (na první pohled je patrný rozdíl mezi slepičím a křepelčím vejcem).

Tato tabulka slouží k zaznamenání snůšky vajec na úrovni lokace. Drobní chovatelé prodávají pouze přebytky není třeba řešit expirace.

Tab. 54: Struktura tabulky 10026 Eggs Locations
(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Eggs_Location	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_Animal_Type	označení druhu zvířete, které vejce sneslo	Integer			ano		ne
ID_Location	označení lokace	Integer			ano		ne
ID_UOM	měrná jednotka	Integer			ano		ne
Quantity	množství vajec	Tinyint					ne
DateTime	datum a čas odběru vajec z lokace	DateTime					ne

3.3.1.47 Tabulka 10027 Eggs_Hatching

Vzhledem ke skutečnosti, že drobní chovatelé mohou provádět líhnutí násadových vajec, je třeba průběh líhnutí a počet vylíhnutých jedinců evidovat.

V průběhu líhnutí jsou prováděny kontroly vajec, které mají za cíl vyřadit vejce, která nebyla oplodněna, případně ta, v nichž se plod nevyvíjí, nebo bylo zjištěno jejich mechanické poškození. Tyto údaje jsou vhodné pro vyhodnocování kvality násadových vajec.

Násadová vejce mohou drobní chovatelé získat buď v rámci vlastního chovu, nebo jejich nákupem.

Tab. 55: Struktura tabulky 10027 Eggs Hatching

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Eggs_Hatch	jednoznačné označení záznamu	Integer		ano		ano	ne
ID_User	označení uživatele	Integer			ano		ne
ID_Animal_Type	označení druhu zvířete, které vejce sneslo	Integer			ano		ne
Start_Date	datum zahájení líhnutí	Date					ne
Expected_End_Date	očekávané datum konce líhnutí	Date					ano
Actual_End_Date	datum skutečného ukončení líhnutí	Date					ano
Eggs_Quantity	počet líhnutých vajec	Tinyint					ne
1st_Control	datum prvního prosvícení vajec	Date					ano
1st_Eliminated_Quantity	počet vyřazených vajec po prvním prosvícení	Tinyint					ano
2nd_Control	datum druhého prosvícení vajec	Date					ano
2nd_Eliminated_Quantity	počet vyřazených vajec po druhém prosvícení	Tinyint					ano
Poultry_Quantity	počet vylíhnutých jedinců	Tinyint					ano

3.3.1.48 Tabulka 10028 Hatched_Animals

Pro vyhodnocení kvality násadových vajec je třeba evidovat jedince, kteří se vylíhli v rámci konkrétního líhnutí vajec. Tato tabulka slouží k provázání jedinců s líhnutím.

Tab. 56: Struktura tabulky 10028 Hatched Animals

(Zdroj: vlastní zpracování)

Název pole	Význam pole	Datový typ	Vel.	PK	FK	AI	Null
ID_Hatched_Animal	jednoznačné označení	Integer		ano		ano	ne
ID_Eggs_Hatch	označení líhnutí	Integer			ano		ne
ID_Animal	označení zvířete	Integer			ano		ne

3.4 ER diagram

Propojení všech tabulek, které byly představeny v přechozí kapitole, je znázorněno na následujícím diagramu.

Obr. 15: ER diagram po dokončení logického návrhu

(Zdroj: vlastní zpracování)

3.5 Fyzický návrh databáze

Tato kapitola obsahuje popis způsobu vytvoření jednotlivých tabulek z předchozí kapitoly, návrh vybraných pohledů a triggeru a jejich realizaci a stručný popis možností zálohování.

3.5.1 Vytvoření jednotlivých tabulek

Následující tabulka představuje pořadí, ve kterém byly vytvořeny výše popsané tabulky. Pořadí tvorby jednotlivých tabulek je optimalizováno tak, aby bylo dodrženo pořadí jejich vytváření s ohledem na jejich provázanost pomocí cizích klíčů. Díky tomuto postupu je možné nastavit cizí klíče již při vytvoření tabulky.

Tab. 57: Pořadí tvorby jednotlivých tabulek
(Zdroj: vlastní zpracování)

Pořadí vytvoření	Pořadí tabulky	Číslo Tabulky	Název Tabulky
1	1	01	Users
	2	2002	Source_Types
	3	2003	Entry_Types
2	4	03	Location_Types
	5	04	Vets
	6	06	Units_Of_Measure
	7	09	Exposition_Types
	8	1005	Feeding_Conditions
	9	2005	Cost_Types
	10	2006	Sales_Types
	11	10002	Animal_Types
	12	10003	Death_Types
	13	10011	Dairy_Quality
	14	10013	Wool_Quality
	15	10014	Kernels
	16	10019	Examination_Types
3	17	10025	Activity_Types
	18	02	Locations

	19	07	Materials
	20	08	Expositions
	21	1004	Feed
	22	10004	Breeding_Types
	23	10005	Breed_Types
	24	10007	Features
	25	10027	Eggs_Hatching
4	26	2001	Location_Inventory_Entries
	27	10001	Animals
	28	10008	Feature_Values
	29	10020	Growth_Table
	30	10026	Eggs_Locations
5	31	05	Vet_Actions
	32	1003	Condition_Specific_Feed
	33	2004	Financial_Entries
	34	10006	Animal_Features
	35	10009	Animal_Locations
	36	10010	Daily_Dairy
	37	10012	Wool_Cutting
	38	10015	Animal_Kernel
	39	10017	Animal_Expositions
	40	10018	Animal_Examinations
	41	10021	Animal_Measurements
	42	10024	Activity_Calendar
	43	10028	Hatched_Animals
6	44	1001	Location_Feeding_Plans
	45	10016	Breeding_Permit
7	46	1002	Location_Feeding_Entries
	47	10022	Litter
8	48	10023	Litter_Animals

Pro to, aby bylo možné výše popsaný postup zajistit, je třeba rozdělit tvorbu tabulek do 8 po sobě jdoucích kroků.

3.5.2 Ukázka vytvoření tabulky

Prvním krokem fyzického návrhu je vytvoření všech navržených tabulek popsaných v logickém návrhu ve zvoleném prostředí, v mém případě prostředí XAMPP.

Jednotlivé tabulky je v prostředí XAMPP možné vytvářet dvěma způsoby. Prvním způsobem je zapsáním SQL příkazu a jeho spuštěním.

Například tabulku Locations lze vytvořit příkazem, který je zobrazený na následujícím obrázku.

```
CREATE TABLE Locations (
ID_Location INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
ID_User INT NOT NULL,
ID_Location_Type INT NOT NULL,
Location_Name varchar(100) NOT NULL,
Usable BOOLEAN NOT NULL,
Quarantine BOOLEAN NOT NULL,
FOREIGN KEY (ID_User) REFERENCES users(ID_User),
FOREIGN KEY (ID_Location_Type) REFERENCES location_types(ID_Location_Type))
```

Obr. 16: Vytvoření tabulky pomocí příkazu SQL

(Zdroj: vlastní zpracování)

Druhým způsobem je použití grafického rozhraní, kdy pro jednotlivé sloupce tabulky stačí zadat jejich názvy, datové typy a případná omezení. V takovémto případě je příkaz na vytvoření tabulky vygenerován na pozadí. Následující obrázek představuje vytvoření tabulky Users tímto způsobem.

Jméno tabulky: Users Přidat 1 column(s) Proveď

Název	Typ	Délka/Množina	Výchozí	Porovnávání	Vlastnosti	Nulový	Klíč	A_I
ID_User Pick from Central Columns	INT		Žádná				PRIMARY PRIMARY	✓
Name Pick from Central Columns	VARCHAR	100	Žádná				---	
Email Pick from Central Columns	VARCHAR	255	Žádná				---	
Telephone Pick from Central Columns	VARCHAR	15	Žádná			✓	---	
Password Pick from Central Columns	VARCHAR	255	Žádná				---	
Registration_DateTin Pick from Central Columns	DATETIME		Žádná				---	
MDL_Sheep Pick from Central Columns	BOOLEAN		Žádná				---	
MDL_Goat Pick from Central Columns	BOOLEAN		Žádná				---	
MDL_Dog Pick from Central Columns	BOOLEAN		Žádná				---	
MDL_Poultry	BOOLEAN		Žádná				---	

Obr. 17: Vytvoření tabulky pomocí grafického rozhraní

(Zdroj: vlastní zpracování)

Po vytvoření všech tabulek byly tyto tabulky naplněny testovacími daty, aby bylo možné demonstrovat možné výstupy z navrhované databáze.

3.5.3 Návrh vybraných pohledů, triggeru a procedury

Následující podkapitoly obsahují návrh několika vybraných pohledů a jejich možný výstup na základě testovacích dat. Navrhované pohledy vychází z analýzy současného stavu, kdy drobní chovatelé specifikovali možné výstupy, které by chtěli z databáze získat. Dále je navržen trigger pro tvorbu záznamů v případě, kdy dojde k zaevidování snůšky vajec.

3.5.3.1 Pohled statistika dojení

Drobní chovatelé, kteří provádí dojení chovaných zvířat, potřebují mít přehled o tom, kolik mléka získají od konkrétního zvířete. Tento pohled nabízí možnost zjistit, kolikrát bylo konkrétní zvíře dojeno, kolik bylo celkově získáno mléka, jaké bylo minimum, maximum a jaké bylo průměrné množství nadojeného mléka, a na základě vypočtených údajů vyhodnocovat tyto hodnoty na úrovni definovaných kvalit mléka.

Tento pohled zobrazuje všechna dojení napříč všemi uživateli, čehož se dá využít pro zrychlení dotazování. Pro snazší identifikaci zvířete je zobrazeno jeho registrační číslo, druh, plemeno a jméno.

Kód pro vytvoření tohoto pohledu je zobrazen na následujícím obrázku.

```
CREATE VIEW dairy_statistics2 AS
SELECT
users.ID_User,
animals.Registration_Number AS `Registrační číslo zvířete`,
animal_types.Animal_Type AS `Druh`,
breed_types.Breed_Type AS `Plemeno`,
animals.Calling_Name AS `Jméno`,
dairy_quality.Dairy_Quality AS `Kvalita mléka`,
COUNT(daily_dairy.ID_Daily_Dairy) AS `Počet dojení`,
MIN(daily_dairy.Milk_Quantity) AS `Minumum`,
AVG(daily_dairy.Milk_Quantity) AS `Průměr`,
MAX(daily_dairy.Milk_Quantity) AS `Maximum`,
SUM(daily_dairy.Milk_Quantity) AS `Celkem`,
units_of_measure.UOM_Name AS `Měrná jednotka`
FROM daily_dairy
INNER JOIN animals ON animals.ID_Animal = daily_dairy.ID_Animal
INNER JOIN dairy_quality ON dairy_quality.ID_Dairy_Quality = daily_dairy.ID_Dairy_Quality
INNER JOIN units_of_measure ON units_of_measure.ID_UOM = daily_dairy.ID_Uom
INNER JOIN users ON users.ID_User = animals.ID_User
INNER JOIN animal_types ON animal_types.ID_Animal_Type = animals.ID_Animal_Type
INNER JOIN breed_types ON animals.ID_Breed_Type = breed_types.ID_Breed_Type
GROUP BY
users.ID_User, animals.Registration_Number, dairy_quality.Dairy_Quality
```

Obr. 18: Vytvoření pohledu statistika dojení

(Zdroj: vlastní zpracování)

Na základě testovacích dat může výstup pro konkrétního uživatele vypadat například jako je zobrazeno na následujícím obrázku. V tomto případě jsou zobrazena data pro uživatele, který má ID 2 a jelikož jsou data vyfiltrována na konkrétního uživatele, není nutné zobrazovat jeho ID.

Registrační číslo zvířete	Druh	Plemeno	Jméno	Kvalita mléka	Počet dojení	Minumum	Průměr	Maximum	Celkem	Měrná jednotka
10010010010	Koza	Bílá krátkosrstá	Líza	Dobrá	14	1.300	1.8117857	2.540	25.365	Litr
10010010010	Koza	Bílá krátkosrstá	Líza	Léčená koza	1	1.200	1.2000000	1.200	1.200	Litr
10010010010	Koza	Bílá krátkosrstá	Líza	Špatná	3	1.300	2.2666667	3.000	6.800	Litr
10010010011	Koza	Anglonubijská koza	Máňa	Dobrá	5	2.320	2.5880000	2.900	12.940	Litr
10010010011	Koza	Anglonubijská koza	Máňa	Léčená koza	12	1.300	2.1885833	2.800	26.263	Litr
10010010011	Koza	Anglonubijská koza	Máňa	Špatná	1	2.140	2.1400000	2.140	2.140	Litr
20010010012	Ovce	Valašská ovce	Vločka	Dobrá	9	1.300	1.5620000	2.100	14.058	Litr
20010010013	Ovce	Valašská ovce	Sněženka	Dobrá	5	1.020	1.4800000	2.200	7.400	Litr
20010010013	Ovce	Valašská ovce	Sněženka	Špatná	4	1.240	1.5950000	2.300	6.380	Litr
20010010014	Ovce	Valašská ovce	Lilie	Špatná	10	0.900	1.2800000	1.620	12.800	Litr

Obr. 19: Výstup pohledu statistika dojení pro uživatele

(Zdroj: vlastní zpracování)

3.5.3.2 Pohled aktuální umístění zvířat

Vzhledem k tomu, že chovatelé mohou chovat zvířata na více místech a umístění zvířat se může v čase měnit (přesun na jinou pastvinu), slouží tento pohled k zobrazení všech aktuálních umístění zvířat napříč všemi uživateli.

Pohled vychází ze skutečnosti, že v rámci navrhovaného řešení může mít zvíře pouze jedno aktuální umístění. Zajištění tohoto stavu bude řešeno na úrovni uživatelského rozhraní a pro označení aktivního záznamu slouží sloupec Active. Tento pohled zobrazuje všechny záznamy, které mají v tomto sloupci hodnotu 1.

```
CREATE VIEW all_active_animal_locations AS
SELECT
users.ID_User,
animal_types.Animal_Type AS `Druh` ,
breed_types.Breed_Type AS `Plemeno`,
animals.Registration_Number `Registrační číslo`,
animals.Calling_Name `Jméno`,
locations.Location_Name `Umístění`,
animal_locations.Start_Date AS `Umístěno od`
FROM animal_locations
INNER JOIN animals ON animals.ID_Animal = animal_locations.ID_Animal
INNER JOIN users ON users.ID_User = animals.ID_User
INNER JOIN animal_types ON animal_types.ID_Animal_Type = animals.ID_Animal_Type
INNER JOIN breed_types ON breed_types.ID_Breed_Type = animals.ID_Breed_Type
INNER JOIN locations ON locations.ID_Location = animal_locations.ID_Location
WHERE animal_locations.Active = 1
ORDER BY
users.ID_User, animals.Registration_Number
```

Obr. 20: Vytvoření pohledu umístění zvířat

(Zdroj: vlastní zpracování)

Jelikož chovatel může chovat více zvířat různých druhů a plemen společně, jsou v rámci tohoto pohledu zobrazeny údaje o druhu a plemenu zvířete, jeho registrační číslo a jméno, název umístění a od kdy je zvíře na daném místě umístěno.

Druh	Plemeno	Registrační číslo	Jméno	Umístění	Umístěno od
Koza	Kamerunská koza	10010010014	Bětka	Chlév 1	2020-04-18
Koza	Kamerunská koza	10010010015	Pepina	Chlév 1	2020-04-18
Koza	Kamerunská koza	10010010016	Matylda	Chlév 1	2020-04-18
Pes	Německý ovčák	111003	Trhač	Kotec pro psa 2	2020-01-09
Pes	Německý ovčák	111004	Amálka	Kotec pro psa 1	2020-02-01
Pes	Německý ovčák	111009	Brita	Kotec pro psa 1	2020-02-06
Pes	Německý ovčák	111010	Black	Kotec pro psa 1	2020-02-06
Pes	Německý ovčák	111011	Bea	Kotec pro psa 1	2020-02-06
Ovce	Suffolkská ovce	20010010015	Pan Beran	Louka u rybníku	2020-12-01
Ovce	Suffolkská ovce	20010010016	Jaro	Louka u rybníku	2020-03-05
Ovce	Suffolkská ovce	20010010017	Léto	Louka u lesa	2020-03-05
Ovce	Suffolkská ovce	20010010018	Podzim	Louka u lesa	2020-03-05
Ovce	Suffolkská ovce	20010010019	Zima	Louka u lesa	2020-03-05
Ovce	Suffolkská ovce	20010010020	Jehně 1 od Jara	Louka u lesa	2020-04-01
Ovce	Suffolkská ovce	20010010021	Jehně 2 od Jara	Louka u lesa	2020-04-01
Ovce	Suffolkská ovce	20010010022	Jehně od Podzimu	Louka u lesa	2020-04-07

Obr. 21: Výstup pohledu umístění zvířat

(Zdroj: vlastní zpracování)

3.5.3.3 Pohled mrtvá zvířata

Tento pohled slouží k zobrazení všech zvířat chovatele, u kterých je v tabulce Animals nastavena ve sloupci Alive hodnota 0. Tohoto pohledu lze například využít při vyhodnocování úmrtnosti jednotlivých zvířat.

V případě, že se jedná o hospodářská zvířata, která byla poražena, jsou zobrazeny i vypočtené hodnoty souhrnné porážkové hmotnosti pro jednotlivá plemena.

Tento pohled byl vytvořen pomocí příkazu zobrazeného na následujícím obrázku.

```sql
CREATE VIEW Dead_Animals AS
SELECT
animals.ID_User AS ID_User,
death_types.Death_Type AS Death_Type,
animal_types.Animal_Type AS Animal_Type,
breed_types.Breed_Type AS Breed_Type,
COUNT(animals.ID_Animal) AS Pocet,
MIN(animals.Death_Weight) AS Minimalni_Vaha,
MAX(animals.Death_Weight) AS Maximalni_Vaha,
SUM(animals.Death_Weight) AS Vaha_Celkem,
AVG (animals.Death_Weight) AS Prumerna_Vaha
FROM animals
INNER JOIN death_types ON death_types.ID_Death_Type = animals.ID_Death_Type
INNER JOIN animal_types ON animal_types.ID_Animal_Type = animals.ID_Animal_Type
INNER JOIN breed_types ON breed_types.ID_Breed_Type = animals.ID_Breed_Type
WHERE
animals.Alive = 0
GROUP BY animals.ID_User,death_types.Death_Type,animal_types.Animal_Type,breed_types.Breed_Type
ORDER BY animals.ID_User,death_types.Death_Type
```

Obr. 22: Vytvoření pohledu mrtvá zvířata

(Zdroj: vlastní zpracování)

Následující obrázek představuje využití výše popsaného pohledu pro jednoho konkrétního testovacího uživatele.

Typ úmrtí	Plemeno	Počet	Minimální váha	Maximální váha	Celková váha	Průměrná váha
Porážka	Šumavanka	6	1.980	2.780	14.453	2.4088333
Porážka	Vlaška	4	2.000	3.000	9.350	2.3375000
Stáří	Bílá krátkosrstá	1	NULL	NULL	NULL	NULL
Stáří	Německý ovčák	1	NULL	NULL	NULL	NULL
Úmrtí při porodu	Bílá krátkosrstá	1	NULL	NULL	NULL	NULL

Obr. 23: Výstup pohledu mrtvá zvířata

(Zdroj: vlastní zpracování)

3.5.3.4 Trigger pro vytvoření zásoby vajec

Tento trigger slouží k automatickému vytvoření záznamu v tabulce Location Inventory Entries, která slouží k evidenci všech zásob, a to v případě, že dojde k vytvoření záznamu v tabulce Eggs Locations, která slouží k evidenci snůšky vajec.

Následující obrázek představuje příkaz jazyka SQL pro vytvoření tohoto triggeru.

```
DELIMITER //
CREATE TRIGGER TR_Egg_Inventory AFTER INSERT ON eggs_locations FOR EACH ROW
BEGIN
DECLARE p_User int;
DECLARE P_Animal_Type int;
DECLARE P_Location int;
DECLARE P_UOM int;
DECLARE P_Quantity int;
SET P_Animal_Type = NEW.ID_Animal_Type;
SET P_Location = NEW.ID_Location;
SET P_UOM = NEW.ID_UOM;
SELECT ID_User INTO P_User FROM animal_types WHERE animal_types.ID_Animal_Type = P_Animal_Type;
SET P_Quantity = NEW.Quantity;
INSERT INTO `location_inventory_entries`(`ID_User`, `ID_Location`, `ID_Entry_Type`, `ID_Source_Type`, `ID_Source_UOM`,
`Number_Source_UOM`, `Remaining_Quantity`, `Usable`, `Open`, `Source_Entry_No`) VALUES (P_User, P_Location, 1, 2, P_UOM,
P_Quantity, P_Quantity, 1, 1, P_Animal_Type);

END; //
DELIMITER ;
```

Obr. 24: Vytvoření triggeru pro naskladnění vajec

(Zdroj: vlastní zpracování)

Tento proces lze vysvětlit pomocí následujícího příkladu. Prvním krokem je, že dojde k vložení dat do tabulky Eggs Location pomocí následujícího příkazu INSERT, který slouží k zaznamenání skutečnosti, že uživatel pro určitý druh zvířete zaevidoval snůšku 120 kusů vajec. Zdrojem tohoto příkazu bude, po vytvoření uživatelského rozhraní, zadání dat uživatelem v rámci uživatelského rozhraní.

```
INSERT INTO `eggs_locations` (`ID_Eggs_Location`, `ID_Animal_Type`, `ID_Location`, `ID_UOM`, `Quantity`, `DateTime`)
VALUES (NULL, '14', '1', '4', '120', '2020-05-13 13:48:02');
```

Obr. 25: Příkaz insert pro otestování triggeru

(Zdroj: vlastní zpracování)

Provedením výše uvedeného příkazu INSERT dojde k vložení záznamu do tabulky Eggs Location, ve struktuře, která je patrná na následujícím obrázku.

ID_Eggs_Location	ID_Animal_Type	ID_Location	ID_UOM	Quantity	DateTime
3	14	1	4	120	2020-05-13 13:48:02

Obr. 26: Záznam vytvořený v Egg Location

(Zdroj: vlastní zpracování)

Zároveň se provedením této akce spustí výše popsaný trigger a dojde k vytvoření odpovídajícího záznamu v tabulce Location Inventory Entries. Vznik nového záznamu lze ověřit příkazem SELECT pomocí filtrování na základě typu původu, kdy v testovacím prostředí odpovídá tabulce Animal Type (na který se eviduje snůška vajec) hodnota 2, konkrétní Animal Type, který vejce snesl má hodnotu 14.

ID	User	Location	Akce	Zdroj	Počet MJ	Zbývá	Datum	Použitelné	Otevřené	Zdroj. položka
11	1	1	1	2	120	120	2020-05-03	1	1	14

Obr. 27: Záznam vytvořený v tabulce Location Inventory Entries

(Zdroj: vlastní zpracování)

Při porovnání hodnot sloupců datum, můžeme vidět, že data si vzájemně neodpovídají, neboť pro zápis do tabulky Location Inventory Entries je vkládána hodnota pro datum pomocí časového razítka, tedy datum, kdy uživatel do systému fyzicky zaevidoval snůšku vajec. Nicméně systémová evidence může být oproti reálnému času posunutá, proto je v tabulce Eggs Location možné zadat reálný čas snůšky.

3.5.3.5 Procedura úmrtnost mláďat v rámci vrhu

Tato procedura slouží k vyhodnocování úmrtnosti mláďat pro uživatelsky definovaný typ úmrtí. Procedura je primárně určena k vyhodnocování porodní úmrtnosti, nicméně ji lze využít pro vyhodnocení všech typů úmrtí vztažených k jednotlivým vrhům.

Vstupním parametrem této procedury je tedy ID typu úmrtí, pro které má být vyhodnocení provedeno. Na základě zadaného typu úmrtí procedura určí, kterému uživateli daný typ úmrtí patří a následně pomocí kurzoru prochází všechny evidované vrhy tohoto uživatele.

Principem tohoto kurzoru je, že prochází jednotlivé záznamy o vrzích a k nalezeným vrhům uživatele dopočítává hodnoty uvedené dále a tyto výstupy včetně identifikace samice ukládá do dočasné tabulky, jejíž obsah je na závěr uživateli vypsán. Pro každý vrh vyhodnocuje, kolik jedinců se v rámci vrhu narodilo, kolik z nich zemřelo na daný typ úmrtí a kolik z nich na daný typ úmrtí nezemřelo. Na základě těchto údajů vypočítá úmrtnost na vrh pro zvolený typ úmrtí.

Vytvoření této procedury je zobrazeno na následujícím obrázku.

```
DELIMITER //
CREATE PROCEDURE FUNGUJ_15 (Type_Of_Death int)
BEGIN
DECLARE P_Finished int;
DECLARE P_Litter int;
DECLARE P_User int;
DECLARE P_Animal int;
DECLARE P_Registration_Number varchar(50);
DECLARE P_Breed_Type varchar(150);
DECLARE P_Animal_Type varchar(50);
DECLARE P_Calling_Name varchar(50);
DECLARE P_Total int;
DECLARE P_Alive int;
DECLARE P_Dead int;
DECLARE P_Mortality decimal(7,4);
SELECT death_types.ID_User INTO P_User FROM death_types WHERE death_types.ID_Death_Type=Type_Of_Death;

BEGIN
DECLARE Browse_Litter  CURSOR FOR
SELECT ID_Litter FROM litter JOIN animals on litter.ID_Animal=animals.ID_Animal WHERE animals.ID_User=P_User;
DECLARE CONTINUE HANDLER
FOR NOT FOUND SET P_Finished = 1;
SET P_Finished = 0;
CREATE TEMPORARY TABLE Temp_Litter (User int, Litter int, Fem_Registration_Number varchar(100), Fem_Breed varchar(100),
 Fem_Animal_Type varchar(100), Fem_Calling_Name varchar(100), Total int, Alive int, Dead int, Mortality decimal(7,4));

OPEN Browse_Litter;
Find_Litter : LOOP
FETCH Browse_Litter INTO P_Litter;
IF P_Finished = 1 THEN
LEAVE Find_Litter;
END IF;
SELECT Animals.ID_Animal INTO P_Animal FROM Animals INNER JOIN litter on animals.ID_Animal = Litter.ID_Animal
WHERE ID_Litter = P_Litter;
SELECT Animals.Registration_Number INTO P_Registration_Number FROM Animals WHERE Animals.ID_Animal = P_Animal;
SELECT Animals.Calling_Name INTO P_Calling_Name FROM Animals WHERE Animals.ID_Animal = P_Animal;
SELECT Breed_Types.Breed_Type INTO P_Breed_Type FROM breed_Types INNER JOIN animals
on animals.ID_Breed_Type = breed_Types.ID_Breed_Type WHERE animals.ID_Animal = P_Animal;
SELECT Animal_Types.Animal_Type INTO P_Animal_Type FROM animal_Types INNER JOIN animals
on animals.ID_Animal_Type = animal_Types.ID_Animal_Type WHERE animals.ID_Animal = P_Animal;
SELECT SUM(CASE WHEN litter_animals.ID_Litter_Animal IS NULL THEN 0 ELSE 1 END) INTO P_Total FROM Litter_Animals
WHERE litter_animals.ID_Litter = P_Litter;
SELECT SUM(CASE WHEN animals.ID_Death_Type = Type_Of_Death THEN 0 ELSE 1 END) INTO P_Alive FROM Litter_Animals
INNER JOIN Animals on Litter_Animals.ID_Animal=Animals.ID_Animal WHERE litter_animals.ID_Litter = P_Litter;
SELECT SUM(CASE WHEN animals.ID_Death_Type = Type_Of_Death THEN 1 ELSE 0 END) INTO P_Dead FROM Litter_Animals
INNER JOIN Animals on Litter_Animals.ID_Animal = Animals.ID_Animal WHERE litter_animals.ID_Litter = P_Litter;
SET P_Mortality = 100.00 * P_Dead / P_Total;
INSERT INTO Temp_Litter (User, Litter, Fem_Registration_Number, Fem_Breed, Fem_Animal_Type, Fem_Calling_Name,
Total, Alive, Dead, Mortality) VALUES (P_User, P_Litter, P_Registration_Number, P_Breed_Type, P_Animal_Type,
P_Calling_Name, P_Total, P_Alive, P_Dead, P_Mortality);
END LOOP Find_Litter;
CLOSE Browse_Litter;

SELECT User as 'Uživatel',
Litter as 'Vrh',
Fem_Animal_Type as 'Druh',
Fem_Breed as 'Plemeno',
Fem_Registration_Number as 'Registrační číslo',
Fem_Calling_Name as 'Jméno',
(CASE WHEN Total is NULL THEN 'Nezadáno' ELSE Total END) as 'Počet mláďat',
(CASE WHEN Alive is NULL THEN 'Nelze vypočítat' ELSE Alive END) as 'Živé',
(CASE WHEN Dead is NULL THEN 'Nelze vypočítat' ELSE Dead END) as 'Mrtvé',
(CASE WHEN Mortality is NULL THEN 'Nelze vypočítat' ELSE Mortality END) as 'Procento úmrtnosti'
FROM Temp_Litter;
END;
END; //
DELIMITER ;
```

Obr. 28: Vytvoření procedury pro vyhodnocení úmrtnosti

(Zdroj: vlastní zpracování)

Procedura je volána příkazem CALL „Jméno procedury“ (vstupní parametry). V tomto případě má procedura jeden vstupní parametr, který se jmenuje Type_Of_Death.

Na následujícím obrázku můžeme vidět výstup z této procedury, který získáme zadáním hodnoty vstupního parametru 16, který v testovací databázi reprezentuje úmrtí při porodu pro uživatele 6.

Uživatel	Vrh	Druh	Plemeno	Registrační číslo	Jméno	Počet mláďat	Živé	Mrtvé	Procento úmrtnosti
6	7	Pes	Dalmatin	111012	Corinka	3	3	0	0.0000
6	8	Pes	Dalmatin	111012	Corinka	4	4	0	0.0000
6	9	Pes	Dalmatin	111012	Corinka	5	4	1	20.0000
6	12	Pes	Český fousek	111013	Dáša	8	6	2	25.0000
6	32	Pes	Český fousek	111013	Dáša	Nezadáno	Nelze vypočítat	Nelze vypočítat	Nelze vypočítat
6	13	Koza	Walliserská koza	10010010023	Ája	3	2	1	33.3333
6	14	Koza	Walliserská koza	10010010023	Ája	2	2	0	0.0000
6	16	Koza	Walliserská koza	10010010023	Ája	2	2	0	0.0000
6	17	Koza	Walliserská koza	10010010023	Ája	1	1	0	0.0000
6	18	Koza	Walliserská koza	10010010023	Ája	2	1	1	50.0000
6	19	Koza	Walliserská koza	10010010024	Bája	2	1	1	50.0000
6	21	Koza	Walliserská koza	10010010024	Bája	3	3	0	0.0000
6	22	Koza	Walliserská koza	10010010024	Bája	1	0	1	100.0000
6	23	Koza	Walliserská koza	10010010025	Cája	2	2	0	0.0000
6	24	Koza	Walliserská koza	10010010025	Cája	2	1	1	50.0000
6	25	Koza	Walliserská koza	10010010026	Dája	1	0	1	100.0000
6	26	Ovce	Šumavská ovce	20010010023	Julie	2	2	0	0.0000
6	27	Ovce	Šumavská ovce	20010010023	Julie	1	1	0	0.0000
6	28	Ovce	Šumavská ovce	20010010024	Alžběta	1	0	1	100.0000
6	29	Ovce	Šumavská ovce	20010010024	Alžběta	2	1	1	50.0000
6	30	Ovce	Šumavská ovce	20010010025	Klára	1	1	0	0.0000
6	31	Ovce	Šumavská ovce	20010010025	Klára	Nezadáno	Nelze vypočítat	Nelze vypočítat	Nelze vypočítat

Obr. 29: Výstup procedury pro vyhodnocení úmrtnosti

(Zdroj: vlastní zpracování)

Na výše uvedeném obrázku můžeme pro vrhy 31 a 32 vidět výstup procedury, který nastane v případě, kdy je vrh uživatelem pouze naplánován, ale ještě nedošlo k narození mláďat, a proto nelze vypočítat uvedené hodnoty.

3.5.4 Zálohy databáze

V rámci tohoto návrhu bylo pro zálohování databáze využíváno možnosti exportovat celou strukturu databáze včetně dat v ní obsažených přímo v rámci rozhraní phpMyAdmin. V případě nasazení řešení do ostrého provozu je možné vytvářet automaticky kopie pravidelně (například každý den), případně využít služeb poskytovatele webhostingu, při jehož výběru je vhodné brát v potaz i možnosti nabízených služeb spojených se zálohováním. Nabídky jednotlivých poskytovatelů se značně liší. V rámci nabízených balíčků se může zejména lišit frekvence provádění záloh, jejich dostupnost pro zákazníka, či finanční náročnost případné obnovy z této zálohy.

3.6 Další kroky před dodáním finálního řešení

Tato práce obsahuje návrh databáze pro drobné chovatele. Výstupem této práce je provedení návrhu databáze, nicméně návrh databáze je pouze částí životního cyklu databáze. Na tento návrh bude navazovat ještě několik dalších fází předtím, než budou mít uživatelé mít k dispozici dokončené komplexní řešení. Tyto kroky jsou schematicky znázorněny pomocí diagramu, který je součástí příloh.

3.7 Přínosy navrhovaného řešení

Větší část přínosů bude možné vyhodnotit až v momentě, kdy bude navrhovaná databáze doplněna o uživatelské rozhraní a bude předána k práci uživatelům, protože pouze subjekty, pro které je navrhované řešení určeno, dokážou vyhodnotit jeho přínosy.

Nicméně mezi přínosy, které lze očekávat po zavedení databázové evidence, lze například zařadit:

- zrychlení a zefektivnění evidence,
- možnost evidovat data při výkonu činnosti (webové rozhraní),
- minimální náklady spojené se změnou evidence,
- řešení parametrizovatelné pro jednotlivé uživatele,
- řešení umožňuje evidenci zásob a finančních toků,
- časová úspora spojená se strukturovaným uložištěm,
- řešení je poměrně snadno rozšiřitelné (přidání modulů pro další druhy zvířat).

ZÁVĚR

Hlavním cílem této práce bylo vytvořit návrh databáze, která bude umožňovat evidenci informací o chovaných jedincích drobným chovatelům z mého okolí, pro které není chov primárním zdrojem příjmů.

V rámci této práce jsem tedy navrhl řešení, které vychází z potřeb těchto chovatelů a umožňuje jim evidenci chovaných jedinců, zásob spojených s chovem (krmiva, chovatelské pomůcky) a finančních transakcí vznikajících chovatelskou činností.

Tuto práci jsem rozčlenil na tři hlavní kapitoly, z nichž první byla zaměřena na teoretická východiska z oblasti návrhu databází. Druhá kapitola obsahuje výstupy z analýzy provedené s chovateli z mého okolí. Z této analýzy vyplynulo, že tito chovatelé se zaměřují na chov psů, koz, ovcí a drůbeže a byly specifikovány základní informace, které je ve spojitosti s chovem těchto zvířat nutné evidovat a možnosti drobných chovatelů spojené s používáním informačních technologií. Z těchto informací jsem vycházel ve třetí části práce, která se zabývala návrhem databáze.

Při zpracování návrhu jsem vycházel z metodologie návrhu databáze, která doporučuje rozčlenit návrh do tří na sebe navazujících kroků. V prvním z těchto kroků jsem vytvořil konceptuální návrh databáze, který obsahoval identifikaci základních entit a jejich vzájemných vztahů. Na tento krok jsem navázal ve druhém kroku, tedy logickém návrhu databáze, kde jsem definoval struktury jednotlivých tabulek a datové typy jednotlivých atributů. Posledním krokem bylo vytvoření fyzického návrhu, kde jsem představil vybrané pohledy, trigger a proceduru.

Pro realizaci řešení jsem se rozhodl využít prostředí XAMPP, které umožňuje jak vytvoření samotné databáze, tak i následný vývoj uživatelského rozhraní.

Na závěr jsem uvedl další postup, který bude navazovat na tento návrh za účelem dodání komplexního řešení drobným chovatelům. Kromě jiného jsem zhodnotil přínosy, které lze po dodání komplexního řešení očekávat. Cíle, které jsem si stanovil v úvodu práce, se mi podařilo splnit.

SEZNAM POUŽITÉ LITERATURY

(1) DRUCKER, Peter Ferdinand. *Postkapitalistická společnost*. 1. vyd. Praha: Management Press, 1993. ISBN 8085603314.

(2) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika*. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 9788024726151.

(3) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi*. 3., aktualizované vydání. Praha: Grada Publishing, 2015. Management v informační společnosti. ISBN 9788024754574.

(4) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009. ISBN 9788025123287.

(5) SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 9788025128787.

(6) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozš. Brno: Akademické nakladatelství CERM, 2010. ISBN 9788021441255.

(7) KROENKE, David a David J. AUER. *Databáze*. 1. vyd. Brno: Computer Press, 2015. ISBN 9788025143520.

(8) KOCH, Miloš a Viktor ONDRÁK. *Informační systémy a technologie*. Vyd. 3. Brno: Akademické nakladatelství CERM, 2008. ISBN 9788021437326.

(9) LAURENČÍK, Marek. *SQL: podrobný průvodce uživatele*. První vydání. Praha: Grada Publishing, 2018. Průvodce (Grada). ISBN 9788027107742.

(10) *Database Normalization* [online]. Norsko: W3schools.in, 2020 [cit. 2020-04-25]. Dostupné z: https://www.w3schools.in/dbms/database-normalization/

(11) KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech*. 1. vyd. Praha: Ekopress, 2012. ISBN 9788086929811.

(12) OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2008. ISBN 9788025117071.

(13) *Přehled datových typů v MySQL* [online]. Česká republika: Martin Šimko, 2007 [cit. 2020-04-25]. Dostupné z: http://programujte.com/clanek/2007052903-prehled-datovych-typu-v-mysql/

(14) *How can I add a Boolean field to MySQL?* [online]. Indie: Arjun Thakur, 2019 [cit. 2020-04-25]. Dostupné z: https://www.tutorialspoint.com/how-can-i-add-a-boolean-field-to-mysql

(15) *What is XAMPP?* [online]. Spojené státy: Apache Friends, 2020 [cit. 2020-05-01]. Dostupné z: https://www.apachefriends.org/index.html

(16) *About Apache Friends* [online]. Spojené státy: Apache Friends, 2020 [cit. 2020-05-01]. Dostupné z: https://www.apachefriends.org/about.html

(17) *XAMPP* [online]. Česká republika: it-slovnik.cz, 2020 [cit. 2020-05-01]. Dostupné z: https://it-slovnik.cz/pojem/xampp

(18) LEISS, Oliver a Jasmin SCHMIDT. *PHP v praxi: pro začátečníky a mírně pokročilé*. 1. vyd. Praha: Grada, 2010. Průvodce (Grada). ISBN 9788024730608.

(19) *The Pros and Cons of 8 Popular Databases* [online]. Švýcarsko: Cody Arsenault, 2017 [cit. 2020-05-15]. Dostupné z: https://www.keycdn.com/blog/popular-databases

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

DBMS	database management system
SŘBD	systém řízení báze dat
1NF	první normální forma
2NF	druhá normální forma
3NF	třetí normální forma
BCNF	Boyce-Coddova normální forma
4NF	čtvrtá normální forma
5NF	pátá normální forma
SQL	structured query language
ANSI	American National Standards Institute
ISO	International Organization for Standardization
DDL	data definition language
DQL	data query language
DML	data manipulation langauge
DCL	data control language
PK	primary key
FK	foreign key
AI	auto-increment
OLAP	online analytical processing
OLTP	online transaction processing
XML	extensible markup language

SEZNAM OBRÁZKŮ

Obr. 1:Vztah mezi informacemi a daty 16
Obr. 2: Vztahy komponent databázového systému 17
Obr. 3: Životní cyklus vývoje databázového systému 20
Obr. 4: Ukázka lineárního datového modelu 23
Obr. 5: Ukázka hierarchického datového modelu 23
Obr. 6: Ukázka síťového datového modelu 24
Obr. 7: Ukázka relačního datového modelu 25
Obr. 8: Ukázka vztahu jedna ku jedné 29
Obr. 9: Ukázka vztahu 1:N 29
Obr. 10: Ukázka vztahu N:M 29
Obr. 11: Ukázka úpravy vztahu N:M 30
Obr. 12: Ukázka objektového datového modelu 31
Obr. 13: Ukázka zjednodušeného stylu 39
Obr. 14: Grafické znázornění vztahů mezi entitami 55
Obr. 15: ER diagram po dokončení logického návrhu 96
Obr. 16: Vytvoření tabulky pomocí příkazu SQL 99
Obr. 17: Vytvoření tabulky pomocí grafického rozhraní 100
Obr. 18: Vytvoření pohledu statistika dojení 101
Obr. 19: Výstup pohledu statistika dojení pro uživatele 102
Obr. 20: Vytvoření pohledu umístění zvířat 102
Obr. 21: Výstup pohledu umístění zvířat 103
Obr. 22: Vytvoření pohledu mrtvá zvířata 104
Obr. 23: Výstup pohledu mrtvá zvířata 104
Obr. 24: Vytvoření triggeru pro naskladnění vajec 105
Obr. 25: Příkaz insert pro otestování triggeru 105
Obr. 26: Záznam vytvořený v Egg Location 105
Obr. 27: Záznam vytvořený v tabulce Location Inventory Entries 106
Obr. 28: Vytvoření procedury pro vyhodnocení úmrtnosti 107
Obr. 29: Výstup procedury pro vyhodnocení úmrtnosti 108

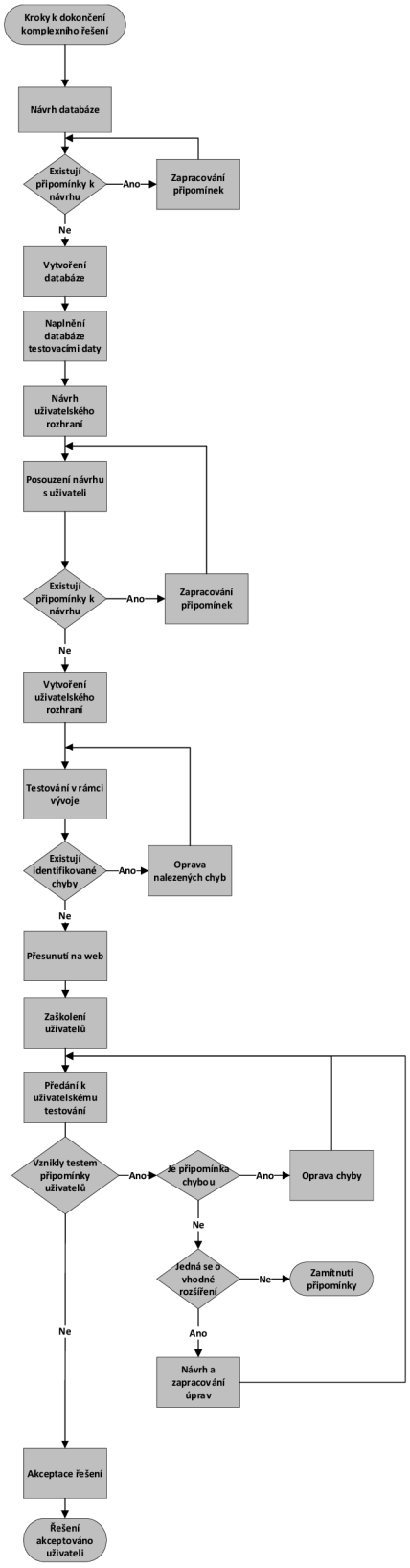
SEZNAM TABULEK

Tab. 1: Restriktivní pravidla pro zajištění referenční integrity 28
Tab. 2: Kroky konceptuálního návrhu databáze 32
Tab. 3: Kroky logického návrhu databáze 33
Tab. 4: Kroky fyzického návrhu databáze 33
Tab. 5: Vybrané datové typy MySQL 38
Tab. 6: Identifikace základních entit 52
Tab. 7: Identifikace základních vztahů 54
Tab. 8: Identifikace atributů základních entit 56
Tab. 9: Struktura tabulky 01 Users 61
Tab. 10: Struktura tabulky 02 Locations 62
Tab. 11: Struktura tabulky 03 Location Types 62
Tab. 12: Struktura tabulky 04 Vets 63
Tab. 13: Struktura tabulky 05 Vet Actions 64
Tab. 14: Struktura tabulky 06 Units Of Measure 64
Tab. 15: Struktura tabulky 07 Materials 65
Tab. 16: Struktura tabulky 08 Expositions 66
Tab. 17: Struktura tabulky 09 Exposition Types 67
Tab. 18: Struktura tabulky 1001 Location Feeding Plans 68
Tab. 19: Struktura tabulky 1002 Location Feeding Entries 69
Tab. 20: Struktura tabulky 1003 Condition Specific Feed 70
Tab. 21: Struktura tabulky 1004 Feed 71
Tab. 22: Struktura tabulky 1005 Feeding Conditions 71
Tab. 23: Struktura tabulky 2001 Location Inventory Entries 72
Tab. 24: Struktura tabulky 2002 Source Types 73
Tab. 25: Struktura tabulky 2003 Entry Types 73
Tab. 26: Struktura tabulky 2004 Financial Entries 74
Tab. 27: Struktura tabulky 2005 Cost Types 75
Tab. 28: Struktura tabulky 2006 Sales Types 75
Tab. 29: Struktura tabulky 10001 Animals 77
Tab. 30: Struktura tabulky 10002 Animal Types 78
Tab. 31: Struktura tabulky 10003 Death Types 79

Tab. 32: Struktura tabulky 10004 Breeding Types 79
Tab. 33: Struktura tabulky 10005 Breed Types 80
Tab. 34: Struktura tabulky 10006 Animal Features 81
Tab. 35: Struktura tabulky 10007 Features 81
Tab. 36: Struktura tabulky 10008 Feature Values 82
Tab. 37: Struktura tabulky 10009 Animal Locations 82
Tab. 38: Struktura tabulky 10010 Daily Dairy 83
Tab. 39: Struktura tabulky 10011 Dairy Quality 83
Tab. 40: Struktura tabulky 10012 Wool Cutting 84
Tab. 41: Struktura tabulky 10015 Wool Quanlity 84
Tab. 42: Struktura tabulky 10014 Kernels 85
Tab. 43: Struktura tabulky 10015 Animal Kernels 86
Tab. 44: Struktura tabulky 10016 Breeding Permits 87
Tab. 45: Struktura tabulky 10017 Animal Expositions 88
Tab. 46: Struktura tabulky 10018 Animal Examinations 88
Tab. 47: Struktura tabulky 10019 Examination Types 89
Tab. 48: Struktura tabulky 10020 Growth Table 90
Tab. 49: Struktura tabulky 10023 Animal Meassurements 90
Tab. 50: Struktura tabulky 10022 Litters 91
Tab. 51: Struktura tabulky 10023 Litter Animals 92
Tab. 52: Struktura tabulky 10024 Activity Calendar 93
Tab. 53: Struktura tabulky 10025 Activity Types 93
Tab. 54: Struktura tabulky 10026 Eggs Locations 94
Tab. 55: Struktura tabulky 10027 Eggs Hatching 95
Tab. 56: Struktura tabulky 10028 Hatched Animals 95
Tab. 57: Pořadí tvorby jednotlivých tabulek 97

SEZNAM PŘÍLOH

Příloha 1: Schematické zobrazení dalšího postupu.....I

Příloha 1: Schematické zobrazení dalšího postupu

(Zdroj: vlastní zpracování)