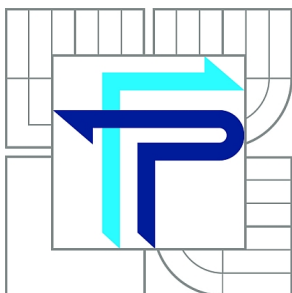


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

DATABÁZE PRO VEDENÍ PLAVECKÉHO ODDÍLU

DATABASE FOR MANAGEMENT OF SWIMMING TEAM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK TÁSLAR

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ KŘÍŽ, Ph.D.

BRNO 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Táslar Marek

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Databáze pro vedení plaveckého oddílu

v anglickém jazyce:

Database for Management of Swimming Team

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

- CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- HERNANDEZ, M. J. Návrh databází. Praha: Grada, 2006. ISBN 80-247-0900-7.
- KOCH, M. a B. NEUWIRTH. Datové a funkční modelování. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.
- LACKO, Ľ. 1001 tipů a triků pro SQL. Brno: Computer Press, 2011. ISBN 978-80-251-3010-0.
- SCHNEIDER, R. D. a L. ROUBÍČEK. MySQL: oficiální průvodce tvorbou, správou a laděním databází. Praha: Grada Publishing, 2006. ISBN 80-247-1516-3.

Vedoucí bakalářské práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 27.05.2013

Abstrakt

Obsahem této práce je analýza požadavků a návrh databáze pro vedení plaveckého oddílu Tesla Brno. Práce se dělí na teoretickou a praktickou část. V teoretické části jsou vysvětleny pojmy a metodologie týkající se návrhu databáze. Praktická část se zaměřuje na analýzu současného stavu, návrhy na zlepšení, návrh databáze a její vytvoření pomocí jazyka SQL.

Abstract

The content of this thesis is the analysis of requirements and database design for Management of the swimming team Tesla Brno. The thesis is divided into theoretical and practical parts. The theoretical part explains the terminology and methodologies related to database design. The practical part focuses on the analysis of the current situation, suggestions for improvement, database design and creating it using SQL.

Klíčová slova

Databáze, Datové modelování, ER diagram, SQL

Keywords

Database, Data modeling, ER diagram, SQL

Bibliografická citace

TÁSLAR, M. *Databáze pro vedení plaveckého oddílu*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 75 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D..

Čestné prohlášení

Prohlašuji, že předložená bakalářská 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 30. května 2013

.....

Marek Táslar

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Jiřímu Křížovi, Ph.D. za jeho cenné rady a pomoc při řešení této práce. Dále bych chtěl poděkovat paní Pavlíně Navrátilové, BA z vedení plaveckého oddílu za ochotu spolupracovat a přístup k informacím.

Obsah

Úvod.....	12
Cíle práce, metody a postupy zpracování	12
1 Teoretická východiska práce	13
1.1 Definice základních pojmů	13
1.1.1 Databázový systém	13
1.1.2 Systém řízení báze dat	13
1.1.3 Databázová aplikace	13
1.1.4 Databáze.....	13
1.1.5 Data.....	13
1.1.6 Informace	13
1.2 Datové modelování	14
1.2.1 Entita.....	14
1.2.2 Datová položka	14
1.2.3 Datový model.....	14
1.2.4 Lineární datový model	14
1.2.5 Objektový datový model.....	14
1.2.6 Relační datový model	15
1.3 Relační model.....	15
1.3.1 Základní pojmy relačního modelu	15
1.3.2 Vlastnosti relačních tabulek.....	15
1.3.3 Doménová integrita.....	15
1.3.4 Entitní integrita	16
1.3.5 Referenční integrita.....	16
1.3.6 Integritní omezení pro vztahy	16

1.3.7	Normalizace	17
1.4	Metodologie návrhu databáze	18
1.4.1	Konceptuální návrh.....	18
1.4.2	Logický návrh	19
1.4.3	Fyzický návrh	19
1.5	Jazyk SQL	19
1.6	Microsoft SQL Server 2008	20
1.6.1	Datové typy SQL Server	20
2	Analýza současného stavu	23
1.7	Základní informace o společnosti	23
1.8	Výuka plavání	25
1.9	Přihlášky do oddílu	26
1.10	Příspěvky	27
1.11	Členské karty	27
1.12	Pomůcky	27
1.13	Závody	28
1.14	Výsledky ze závodů.....	28
1.15	Trenéři	29
1.16	Docházka	29
1.17	Kniha zranění a ošetřování	30
1.18	Tréninkový deník.....	30
1.19	Plavecký trénink	30
1.20	Další aktivity	31
1.21	Shrnutí požadavků	31
3	Vlastní návrhy řešení	32
3.1	Konceptuální návrh	32

3.1.1	Identifikace entit	32
3.1.2	Identifikace vztahů mezi entitami	33
3.1.3	Identifikace atributů, určení klíčů	33
3.1.4	Clen	34
3.1.5	Trener	36
3.1.6	Kvalifikace	37
3.1.7	Platba	37
3.1.8	ClenskaKarta	38
3.1.9	TreninkovaSkupina	38
3.1.10	Trenink	39
3.1.11	CastTreninku	39
3.1.12	Aktivita	40
3.1.13	ZaznamOZraneni	41
3.1.14	Zavod	41
3.1.15	Vysledek	42
3.1.16	Pomucka	43
3.1.17	Dodavatel	43
3.1.18	ZaznamOPoskozeni	44
3.2	Logický návrh	46
3.2.1	ClenskaKarta	46
3.2.2	Platba	46
3.2.3	Trener	47
3.2.4	Kvalifikace	47
3.2.5	TreninkovaSkupina	48
3.2.6	Trenink	49
3.2.7	CastTreninku	49

3.2.8	Aktivita	50
3.2.9	ZaznamOZraneni	51
3.2.10	Vysledek	51
3.2.11	Zavod	53
3.2.12	ZaznamOPoskozeni	53
3.2.13	Pomucka.....	54
3.2.14	Dodavatel	54
3.2.15	Clen.....	54
3.3	Fyzický návrh.....	55
3.3.1	Procedura VlozClena	55
3.3.2	Procedura CelkemZaObdobi.....	56
3.3.3	Procedura CelkemZaObdobiClen	57
4	Závěr	58
	Seznam literatury	59
	Seznam obrázků.....	60
	Seznam tabulek.....	60
	Seznam příloh	61

Úvod

Tělovýchovná jednota Tesla Brno je sdružení zabývající se již mnoho let provozem sportovních zařízení a činností sportovních oddílů. Cílem organizace je nabídka sportovního vyžití pro širokou veřejnost. Plavecký oddíl je jednou z činností, kterou toto sdružení provozuje téměř od začátku svého působení.

Plavecký oddíl má v současné době desítku zaměstnanců a zhruba 150 členů. Cílem plaveckého oddílu je nabídnout zájemcům o tento druh sportu kvalitní výuku plavání prostřednictvím svých zaměstnanců a dostupných pomůcek.

Vedení takového oddílu potřebuje vést záznamy o svých členech, majetku a provozovaných aktivitách. V dnešní době, plně dostupných informačních technologií, se zdá být používání papírových dokumentů pro veškeré záznamy jako zastaralý způsob uchování dat. Papírová forma záznamu dat má také omezené možnosti přístupu k těmto informacím a jejich sdílení s veřejností. Analýza těchto dat je bez použití výpočetní techniky pracná a časově velmi náročná.

Z těchto důvodů vznikl požadavek na vytvoření databázového systému, který umožní organizaci ukládat potřebná data a usnadní přístup k nim.

Cíle práce, metody a postupy zpracování

Hlavním cílem této práce je navrhnout organizaci databázi se strukturou umožňující uložit veškeré záznamy, které organizace ukládat potřebuje. K tomu využívám metody datového modelování a znalosti databázových systémů a především jazyka SQL. Požadavky a potřeby organizace jsou shrnuty v kapitole Analýza současného stavu. Z těchto požadavků sestavuji nejprve konceptuální návrh, dále logický návrh, podle kterého nakonec vytvořím databázi pomocí příkazů jazyka SQL.

1 Teoretická východiska práce

1.1 Definice základních pojmů

1.1.1 Databázový systém

Pod pojmem databázový systém rozumíme kolekci databázových aplikací, systému řízení báze dat (DBMS) a samotnou databázi. (1)

1.1.2 Systém řízení báze dat

Systém řízení báze dat (DBMS – Database Management System) je software, který je propojujícím článkem mezi databázovými aplikacemi a databází. Umožňuje uživatelům práci s daty v databázi. Zejména jejich vkládání, aktualizaci, mazání a vyvolání. Dotazování na tato data většinou probíhá pomocí dotazovacího jazyka. (1)

1.1.3 Databázová aplikace

„Počítačový program interagující s databází vyvoláním odpovídajícího požadavku (typicky jednoho nebo více příkazů SQL) pro DBMS“ (Conolly, 2009, str. 39).

Databázovou aplikaci používají uživatelé k vytváření a správě databáze a generování informací. V současné době se velmi často jedná o online aplikaci. (1)

1.1.4 Databáze

Databázi můžeme chápat jako seskupení vzájemně souvisejících dat, se kterými pracujeme jako s ucelenou jednotkou. Podrobnější definice záleží na produktu konkrétního výrobce. Například v produktu Microsoft Access se celá databáze ukládá do jednoho souboru obsahujícího příslušné datové položky a společnost Oracle definuje databázi jako kolekci několika souborů. (6)

1.1.5 Data

„Surová (nezpracovaná) fakta, která mají určitou důležitost pro jednotlivce nebo organizaci“ (Conolly, 2009, str. 36).

1.1.6 Informace

„Data, která prošla zpracováním nebo dostala strukturu, která jim dává pro jednotlivce nebo organizaci význam“ (Conolly, 2009, str. 36).

1.2 Datové modelování

1.2.1 Entita

Entita je datový objekt, který reprezentuje každý reálný datový objekt. Pro každý datový objekt musíme definovat údaje, které chceme o reálném objektu uchovávat. Těmto atributům říkáme atributy entity. Například pro entitu Automobil mohou být atributy Výrobce, Model, Obsah motoru, SPZ. (3)

1.2.2 Datová položka

Atomický (dále nedělitelný) údaj, který je charakterizován typem a délkou. Nejzákladnějšími a nejčastěji používanými datovými položkami jsou text, číslo a datum. Konkrétní typy datových položek závisí na použitém DBMS.

1.2.3 Datový model

„Datový model je architektura, podle které databázový systém ukládá objekty do databáze a podle které je vzájemně provazuje“ (Oppel, 2006, str. 22).

Nejzákladnějšími typy datových modelů jsou:

- lineární,
- relační,
- objektový.(3)

1.2.4 Lineární datový model

Lineární datový model nemá žádné informace o struktuře souboru nebo jakékoliv vztahy mezi záznamy. Informace o struktuře dat a jejich významu musí obsahovat příslušná aplikace nebo je musí znát cílový uživatel. Tyto nevýhody vedly k vytvoření vhodnějších datových modelů. (6)

Výhodou tohoto modelu je, že ho jako jediný model můžeme implementovat na libovolném médiu (například papírová kartotéka pacientů, kde jednotlivé karty jsou uloženy v krabici). (3)

1.2.5 Objektový datový model

Základním prvkem tohoto datového modelu je objekt, který má kromě atributů definovány ještě metody. Objekty, které jsou stejného typu, tvoří třídu objektů. Každý objekt je jednoznačně určen pomocí unikátního identifikátoru. K atributům můžeme

přístupovat pouze voláním definovaných metod. Tomu říkáme zapouzdření objektu a díky této vlastnosti má objektový model vysokou datovou abstrakci a nezávislost dat. (3)

1.2.6 Relační datový model

Jedná se v podstatě o několik lineárních modelů spojených pomocí datové položky či položek, které se nazývají relační klíče. Spojení nejsou trvalá, ale vznikají v případě potřeby získání dat z více tabulek. Tato spojení po práci s modelem opět zanikají. Relační datový model patří k nejpoužívanějším v současnosti. (3)

1.3 Relační model

1.3.1 Základní pojmy relačního modelu

Relace (entity) jsou v relačním modelu reprezentovány tabulkami. Tato tabulka je dvourozměrná - má sloupce a řádky. Každý pojmenovaný sloupec relace se nazývá atribut. Řádek relace nazýváme datová n-tice. Doménou rozumíme množinu hodnot, které jsou pro daný atribut přípustné. Relační databáze je kolekce normalizovaných tabulek. (1)

1.3.2 Vlastnosti relačních tabulek

- Každá tabulka má jednoznačné jméno, kterým se odlišuje od ostatních tabulek stejné databáze.
- Každá buňka tabulky smí obsahovat pouze jednu hodnotu.
- Každý sloupec tabulky má jedinečné jméno.
- Hodnoty jednoho atributu jsou ze stejné domény
- Pořadí sloupců je nevýznamné
- Každý záznam je jednoznačný
- Pořadí záznamů nemá význam (1)

1.3.3 Doménová integrita

Toto integritní omezení se týká hodnot. Říká nám, že každá hodnota atributu relace musí být z množiny hodnot pro tento atribut přípustných. Doménu specifikujeme jako množinu hodnot nebo specifikujeme povolené hodnoty pomocí:

- datového typu,

- povinnosti zadat či nezadat hodnotu položky (NULL),
- jedinečnosti hodnoty v rámci sloupce,
- rozsahu hodnot,
- implicitní hodnoty,
- masky pro vkládání,
- seznamu přípustných hodnot (číselníku). (3)

1.3.4 Entitní integrita

Entitní integrita se týká primárního klíče. **Primární klíč** vybíráme z klíče kandidátního. **Kandidátní klíč** je jeden či více atributů, které jednoznačně určují řádek tabulky. Klíč, který si nakonec vybereme, se stane klíčem primárním, ostatní klíče budou klíči alternativními.

Vlastnosti primárního klíče jsou:

- jedinečnost – v jedné tabulce neexistuje n-tice, která by měla stejnou hodnotu primárního klíče,
- neredukovatelnost (klíč je minimální) – nemůžeme vypustit žádný z atributů primárního klíče, aniž bychom porušili jednoznačnost. (3)

1.3.5 Referenční integrita

Referenční integrita souvisí s pojmem cizí klíč. **Cizí klíč** je atribut či skupina atributů, které odpovídají kandidátnímu klíči některé z tabulek. (1)

Další vlastností cizího klíče je, že jeho hodnota je buď plně zadaná či plně nezadaná. Právě cizí klíč spolu s odpovídajícím kandidátním klíčem jiné tabulky nám umožňuje spojovat tabulky, tedy hlavní účel relačního modelu. (3)

1.3.6 Integritní omezení pro vztahy

Obecně máme čtyři typy vztahů mezi jednotlivými relacemi.

Prvním typem je **vztah 1:1**. Ten nám říká, že jeden záznam v tabulce odpovídá právě jednomu záznamu v jiné tabulce.

Příkladem takového vztahu by mohl být vztah entit člověk a občanský průkaz. Kde platí, že jeden člověk (dospělý) vlastní jeden občanský průkaz. Zároveň jeden občanský průkaz je vlastněn pouze jedním člověkem.

Dalším typem je **vztah 1:N**. Tento vztah nám říká, že jednomu záznamu v tabulce odpovídá více záznamům v jiné tabulce. Příkladem by mohl být vztah mezi zákazníkem a objednávkou. Jeden zákazník vytváří obvykle více objednávek, ale jedna konkrétní objednávka náleží právě jednomu konkrétnímu zákazníkovi.

Vztah N:1 je stejný jako vztah 1:N, pouze se na něj pohlížíme z druhé strany.

Vztah N:M je dalším typem. Tento vztah nám říká, že více záznamům v tabulce odpovídá více záznamům jiné tabulky. Příkladem je případ vztahu mezi předmětem a studentem. Platí, že jeden student studuje více předmětů a jeden předmět je studován více studenty. (3)

Zvláštním typem je **rekurzivní vztah**. Tento vztah je mezi dvěma instancemi stejné entity. Pokud například v tabulce o zaměstnancích budeme sledovat nadřízeného pomocí atributů - číslo zaměstnance a číslo nadřízeného, může to znamenat, že jeden zaměstnanec je nadřízeným jednoho či více zaměstnanců. (6)

1.3.7 Normalizace

„Technika používaná pro vytvoření sady tabulek s minimální redundancí, která podporuje datové požadavky organizace“ (Conolly, 2009, str. 188).

1. normální forma (1NF)

Tabulka splňuje podmínky první normální formy, pokud každý průsečík sloupce a záznamu obsahuje právě jednu jedinou hodnotu. (1)

Atributy tedy nesmí být složené či vícehodnotové. Příkladem složeného atributu by mohla být adresa zapsaná v jedné buňce tabulky. Abychom toto porušení odstranili, rozložíme atribut Adresa, na atributy Ulice, Číslo popisné, Obec, PSČ. (3)

2. normální forma (2NF)

„Tabulka, která je v 1NF a ve které jsou hodnoty každého sloupce, který není součástí primárního klíče, determinovány všemi hodnotami sloupců, které tvoří primární klíč“ (Conolly, 2009, str. 192).

Tato normální forma se týká pouze tabulek, které obsahují složený primární klíč. Z toho plyne, že pokud tabulka obsahuje primární klíč tvořený pouze jedním sloupcem, je v 2NF automaticky. 2NF se týká funkční závislosti. Pokud je v navržené tabulce pouze částečná závislost na primárním klíči, musíme ji odstranit. To provedeme odstraněním

částečně závislých sloupců z původní tabulky a jejich umístěním do nové tabulky. Na hodnoty sloupců v této nové tabulce se pak odkazujeme pomocí cizího klíče umístěného v původní tabulce. (1)

3. normální forma

„Relace je v třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé“ (Koch, 2010, str. 58).

Tabulka ve třetí normální formě nesmí obsahovat tranzitivní závislosti. Každý atribut, který není součástí primárního klíče, musí být funkčně závislý na celém klíči. Pokud je tato závislost zprostředkována pomocí atributu, který není klíčový, je tranzitivně závislý. Abychom tedy dostali tabulku do 3NF, musíme provést její dekompozici a tranzitivně závislé atributy umístit do nové tabulky. Cizím klíčem v původní tabulce se pak budeme odkazovat na primární klíč nové tabulky. (3)

Dále se ještě můžeme setkat s Boyce – Coddovou, 4. a 5. normální formou. Ty ale nejsou tak časté a jejich použití vyžadují některé speciální případy. Obecně můžeme říci, že pro korektní fungování databáze si vystačíme s prvními třemi normálními formami. (3)

1.4 Metodologie návrhu databáze

Metodologie návrhu databáze se skládá ze tří hlavních částí:

- Konceptuální návrh
- Logický návrh
- Fyzický návrh

1.4.1 Konceptuální návrh

V tomto kroku se snažíme vytvořit model dat, které používá organizace, ale neuvažujeme ještě finální implementaci. Cílem konceptuálního návrhu je vytvoření ER modelu. ER model se skládá z ER modelu a slovníku dat.

Postup při konceptuálním návrhu je určen následujícími kroky:

- Identifikace entit.
- Identifikace vztahů mezi entitami.

- Identifikace atributů jednotlivých entit.
- Určení domén atributů.
- Určení kandidátních, primárních a alternativních klíčů.
- Specializace/generalizace entit
- Kontrola redundance
- Kontrola uživatelských transakcí
- Posouzení návrhu s uživateli databáze (1)

1.4.2 Logický návrh

Logický návrh vychází z návrhu konceptuálního. Entity zachycené v ER modelu reprezentujeme pomocí tabulek. V každé tabulce jsou vyznačeny všechny atributy, primární klíče a také cizí klíče potřebné pro vytvoření spojení mezi tabulkami.

Kroky logického návrhu jsou:

- Vytvoření tabulek.
- Kontrola vytvořených tabulek pomocí normalizace.
- Kontrola uživatelských transakcí.
- Kontrola integritních omezení.
- Posouzení s uživateli databáze. (1)

1.4.3 Fyzický návrh

Tato část návrhu databáze se týká finální implementace uzpůsobené pro konkrétní DBMS. Normalizované tabulky z logického návrhu fyzicky vytvoříme pomocí příkazů, které nám cílový DBMS umožňuje. Ve většině případů se jedná o jazyk SQL.

1.5 Jazyk SQL

Jazyk SQL (Structured Query Language - strukturovaný dotazovací jazyk) je standardním jazykem pro práci s daty v relační databázi. Pomocí jazyka SQL může člověk pracující s databází provádět:

- Úpravy struktury databáze
- Změny nastavení zabezpečení systému
- Přidávat uživatelská oprávnění k databázím či tabulkám

- Dotazy nad tabulkami databáze
- Aktualizace obsahu tabulek (7)

Nejčastějším příkazem jazyka SQL je příkaz SELECT. Ten slouží k získávání dat z databáze a vrací je uživateli. Dalšími typickými příkazy jsou:

- CREATE – slouží k vytváření objektů databáze, zejména tabulek
- INSERT – vkládá data do databáze
- UPDATE – aktualizuje data v databázi
- DELETE – maže záznamy v databázi

Nejpopulárnějšími implementacemi jazyka SQL jsou MySQL, Oracle Database, Microsoft SQL Server, Sybase a IBM DB2. (7)

K důležitým vlastnostem SQL patří podpora procedur uložených na databázovém serveru. Tím, že se provádějí přímo na serveru, mají pozitivní vliv na rychlost databáze. (4)

1.6 Microsoft SQL Server 2008

Produkt SQL Server je řešením společnosti Microsoft a nabízí komplexní databázový systém včetně pokročilých funkcí (od analytických nástrojů po dolování dat), které jsou závislé na použité verzi. (5)

1.6.1 Datové typy SQL Server

Datové typy lze rozdělit na číselné, znakové, datové a časové, binární a ostatní typy.

Mezi ty **číselné** patří:

- **tinyint** – celočíselná hodnota o rozsahu 0 až 255 (velikost 1 bajt)
- **smallint** – celočíselná hodnota o rozsahu -32768 až 32767 (2 bajty)
- **int** – celočíselná hodnota o rozsahu -2^{31} až $2^{31}-1$ (4 bajty)
- **bigint** – celočíselná hodnota o rozsahu -2^{63} až $2^{63}-1$ (8 bajtů)
- **decimal(p,s)** – číselná hodnota s pevnou desetinnou čárkou o rozsahu $-10^{38}+1$ až $10^{38}-1$ (5-17 bajtů); p – značí celkový počet číslic, s - počet číslic za desetinnou čárkou

- **numeric(p,s)** – rovnost s datovým typem decimal, je zachován kvůli kompatibilitě
- **smallmoney** – peněžní hodnoty o rozsahu -214748.3648 až 214748.3647 (4 bajty)
- **money** – peněžní hodnoty o rozsahu -922337203685477.5808 až 922337203685477.5807 (8 bajtů)
- **real** – číselná hodnota s plovoucí desetinnou čárkou o rozsahu -3.4^{38} až -1.18^{38} , 0 a 1.18^{38} až 3.4^{38} (4 bajty)
- **float(n)** – plovoucí desetinná čárka o rozsahu -1.79^{308} až -2.23^{308} , 0 a 2.23^{308} až 1.79^{308} (4 až 8 bajtů)

Mezi **znakové** datové typy patří:

- **char(n)** – znakový řetězec pevné délky o velikosti 1 bajt na znak (1 bajt až 8000 bajtů)
- **varchar(n)** – znakový řetězec proměnlivé délky o velikosti 1 bajt na znak (1 bajt až 8000 bajtů)
- **text** – znakový řetězec o proměnlivé délce a velikosti až 2GB; jedná se o zastaralý datový typ, který by neměl být používán
- **nchar()** – znakový řetězec pevné délky o velikosti 2 bajty na znak (2 bajty až 4000 bajtů); kódování Unicode
- **nvarchar** – znakový řetězec proměnlivé délky o velikosti 2 bajty na znak (2 až 4000 bajtů); kódování Unicode
- **ntext** – to samé jako text, ovšem pro kódování Unicode a velikost je 2 bajty na znak

Pro práci s datem a časem nabízí SQL Server tyto datové typy:

- **smalldatetime** – rozsah 01/01/1900 až 06/06/2079, přesnost 1 minuta (4 bajty)
- **datetime** – rozsah 01/01/1753 až 12/31/9999, přesnost 0.00333 sekundy (8 bajtů)
- **datetime2** – 01/01/0001 až 12/31/9999, přesnost 100 nanosekund (6-8 bajtů)
- **datetimeoffset** – rozsah a přesnost jako datetime, umožňuje navíc ukládat časové pásmo (8-10 bajtů)
- **date** – rozsah 01/01/0001 až 12/31/9999, přesnost 1 den (3 bajty)

- **time** – rozsah 00:00:00.0000000 až 23:59:59.9999999, přesnost 100 nanosekund (3-5 bajtů)

Binární data:

- **bit** – nabývá pouze tří hodnot – NULL, 0 nebo 1 (velikost 1 bit)
- **binary** – binární data s pevnou šířkou (až 8000 bajtů)
- **varbinary** – binární data s proměnlivou šířkou (až 8000 bajtů)
- **image** – binární data s proměnlivou šířkou (až 2 GB), zastaralý typ, který může být nahrazen typem varbinary(max)

Ostatní datové typy:

- XML – umožňuje ukládat dokumenty XML (eXtensible Markup Language) až do velikosti 2GB a 128 úrovní.
- Data typu FILESTREAM pro data typu BLOB
- Prostorový datový typ – geometry (body, čáry, křivky, polygony) a geography (zeměpisná šířka a délka) (2)

2 Analýza současného stavu

V této kapitole nejprve představím společnost a její hlavní činnost. Poté se zaměřím na oddíl plavání, který je předmětem této práce. Popíši situaci v oddíle z pohledu vedení a pokusím se ji zanalyzovat, z čehož následně vyvodím návrhy na zlepšení.

1.7 Základní informace o společnosti

TĚLOVÝCHOVNÁ JEDNOTA TESLA BRNO

- Halasovo nám. 7, 638 00 Brno-Lesná
- Tel.: 545 222 600
- IČO: 00214086
- DIČ: CZ00214086
- Právní forma: Sdružení (svaz, spolek, společnost, klub aj.)
- Datum vzniku: 16. květen 1990
- Počet zaměstnanců: 25 - 49 zaměstnanců

Předmětem činnosti, dle Českého statistického úřadu, je provozování sportovních zařízení. Tuto činnost provozuje společnost především v hlavním areálu TJ Tesla v Brně na Halasově náměstí, kde jsou k dispozici tyto sportoviště a služby:

- Plavecký bazén (25m)
- Sportovní hala
- Gymnastický sál
- Posilovna
- Sauna
- Venkovní brouzdaliště
- Klubovna
- Bufet.

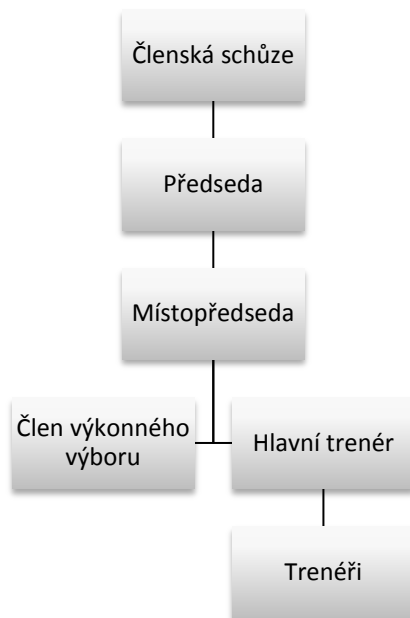
Další sportoviště – loděnice – se nachází v městské části Komín a posledním areálem je turistická chata v Jeseníkách. Společnost zajišťuje provoz zařízení především pro sportovní oddíly, ale i pro neorganizovanou veřejnost.

Sportovními kluby působícími pod hlavičkou TJ Tesla jsou:

- Basketbal
- Horolezectví
- Kanoistika
- Karate
- Klub orientačních sportů
- Kulturistika
- Lyžování
- Moderní gymnastika
- Plavání
- Synchronizované plavání
- Turistika.

Členové jednotlivých oddílů provozují jejich sportovní činnost od městské až po reprezentační úroveň. (9)

Základní organizační jednotkou TJ Tesla Brno je oddíl. Oddíl si utváří vlastní organizační strukturu. Nejvyšším orgánem oddílu plavání je členská schůze. Na členské schůzi se volí předseda a místopředseda oddílu. Dále se určí členové výkonného výboru a hlavní trenér. Tyto funkce jsou většinou voleny na čtyři roky. Členové výkonného výboru jsou tři osoby – předseda, místopředseda a jeden zvolený člen oddílu. (8)



Obrázek 1: Organizační struktura vedení plaveckého oddílu (Zdroj: Vlastní zpracování)

1.8 Výuka plavání

Plavecký oddíl má v současné době přibližně 150 členů. Z toho převážnou většinu tvoří děti školního a předškolního věku. Oddíl však nabízí plaveckou výuku i pro mládež a dospělé. Plavci jsou rozděleni podle věku a výkonnosti do jednotlivých výukových hodin a na příslušné plavecké dráhy. K dispozici jsou tyto skupiny:

- plavání pro neplavce a začátečníky - děti 4-9 let
- kondiční plavání pro děti 10-15 let
- závodní plavání pro děti od 8 let
- kondiční a zdokonalovací plavání pro dospělé

Pro děti předškolního věku a začátečníky jsou v současné době určeny hodiny v úterý a čtvrtek, kdy tyto děti plavou přibližně 30 minut na jedné polovině plavecké dráhy. Tato výuka je zaměřená spíše na seznámení s vodou, odbourávání strachu z vody a základní pohyby ve vodě. Výuka probíhá především formou různých her. Na druhé polovině dráhy probíhá trénink o něco starších a zdatnějších dětí. Tyto děti většinou zvládají pohyb ve vodě bez nadlehčovacích pomůcek. Výuka zde také probíhá za pomoci her, ale v mnohem menší míře. Tato výuka trvá hodinu. Pondělí je určeno pro závodní plavání a kondiční plavání zdatnějších dětí. Tento trénink trvá přibližně 60 minut pro obě skupiny. U závodního plavání je prioritou naučit děti plavat všemi plaveckými způsoby a připravit je na plavecké závody. Ve středu je první hodina věnována pro kondiční plavání dětí od 8 do 12 let a druhá hodina pro děti až do věku 15 let. Další ze čtvrtečních hodin je určena pro plavání dospělých a závodní plavání. Pro přehlednost ještě uvádím tabulku s jednotlivými výukovými hodinami a příslušnými skupinami.

Tabulka 1: Výuka plavání v TJ Tesla Brno k 1.1.2013 (Zdroj: vlastní zpracování)

Den	Čas	Skupina	Věk (orientačně)
Pondělí	16:00-17:00	Kondiční plavání	6-8 let
	17:00-18:00	Kondiční plavání	6-10 let
	17:00-18:00	Závodní plavání	6-12 let
Úterý	17:00-17:30	Neplavci	4-6 let
	17:30-18:00	Začátečníci	4-7 let
	17:00-18:00	Kondiční plavání	6-8 let
Středa	16:00-17:00	Kondiční plavání	8-12 let
	17:00-18:00	Kondiční plavání	12-15 let
Čtvrtek	17:00-17:30	Neplavci	4-6 let
	17:30-18:00	Začátečníci	4-7 let
	17:00-18:00	Závodní plavání	6-12 let
	17:00-18:00	Kondiční plavání dospělých	15+ let

1.9 Přihlášky do oddílu

Aby se někdo mohl stát členem oddílu, musí vyplnit papírové přihlášky. Jedna je přihláška do tělovýchovné jednoty a druhá do oddílu plavání. V přihláškách se mimo jména, příjmení, bydliště a kontaktních údajů, jako jsou telefon a e-mailová adresa, vyplňuje jméno a příjmení rodiče a souhlas s podmínkami plaveckého oddílu, dále vybírá z možností zasílání informací o činnosti oddílu a uveřejněním fotek na webových stránkách oddílu. V neposlední řadě zákonný zástupce svým podpisem stvrzuje správnost údajů a souhlas se zpracováním a evidencí osobních údajů a souhlas s využitím rodného čísla. Osobní údaje oddíl potřebuje především k registraci člena na příslušném sportovním svazu. K tomu dochází hlavně u plavců, kteří trénují v závodním družstvu a jiných členů, které mají zájem účastnit se závodů.

1.10 Příspěvky

Po vyplnění přihlášky následuje platba členských příspěvků. Příspěvky se platí jednorázově na každé pololetí. Po zaplacení je člen oprávněn navštěvovat danou hodinu, do které je přihlášen po celé pololetí. Výjimku tvoří plavání dospělých, kteří si kupují permanentky s příslušným počtem vstupů. Platba probíhá hotovostně přímo v areálu TJ Tesla Brno, kde příslušnou částku vybírá pověřená osoba, která vypíše doklad o převzetí platby. Do budoucna se uvažuje o platbě převodem na společný bankovní účet tělovýchovné jednoty. K identifikaci platby by měl sloužit variabilní symbol skládající se z dvoumístného kódu oddílu, čísla člena a čísla určujícího účel platby.

1.11 Členské karty

Po splnění předchozích kroků je každému členovi vyplněna karta, kterou se prokazuje při vstupu do šaten bazénu. Na kartě je vyplněno jméno a jsou na ní vyznačené hodiny, které je držitel karty oprávněn navštěvovat. Plavci, kteří navštěvují plavání dospělých, mají k této kartě ještě permanentku s určitým počtem hodin, které si zaplatili. Možnostmi jsou permanentka na 5, 10 či 20 hodin.

1.12 Pomůcky

Oddíl zajišťuje svým členům možnosti využití různých plaveckých pomůcek a vybavení pro výuku plavání. Mezi tyto pomůcky patří zejména veškeré plovoucí desky, pontony, hadice, tzv. piškoty, podložky a nadlehčovací pásy. Dále různé plovoucí a potápějící se výrobky určené pro hry ve vodě. Většinu těchto pomůcek má oddíl v době, kdy je nepoužívá, uzamčené ve skladu. Na téměř všech pomůckách má oddíl vyznačeno lihovým fixem zkratku TJTB, aby nedošlo k záměně s pomůckami jiných oddílů.

Objednávání pomůcek má na starosti místopředseda oddílu, který na základě žádosti od trenérů obstará nové pomůcky. Zatím v oddíle neexistují žádné záznamy o dodavatelích tohoto materiálu. Záznamy mají význam zejména při objednávání pomůcek v budoucnosti, kdy již nebude nutné znovu vyhledávat prodejce, ale bude možné získat kontakty z databáze.

V souvislosti se správou tohoto majetku navrhuji zřídit záznamy o závadách či poškození těchto pomůcek. Tyto záznamy mohou sloužit při rozhodování o nakoupení nových pomůcek.

1.13 Závody

Všichni plavci, kteří se chtějí účastnit závodů, které jsou vypsány na termínové listině Českého svazu plaveckých sportů, musí být na daném svazu zaregistrováni a mít zaplacený roční příspěvek. Svaz pak příslušnému plavci vystaví registrační kartu, kterou zašle oddílu, ve kterém je registrován.

Závody se konají po celý rok a jsou určené pro vypsané kategorie. Přihlašování na závody probíhá buď on-line na stránkách svazu (závody M-ČR), případně podle pokynů pořadatele e-mailem či poštou.

1.14 Výsledky ze závodů

Výsledky závodů, jež jsou zaregistrovány na ČSPS, jsou zpracovány na počítači organizátorem akce a poté nahrány na webové stránky plaveckého svazu. Odtud je možné jejich stažení ve formátu PDF či jejich export do textového souboru, kde jsou jednotlivá data oddělena středníkem. To usnadňuje jejich import například do tabulkového procesoru (např. MS Excel). Výsledky ze závodů, které nejsou zaregistrovány na svazu, jsou zasílány většinou e-mailem ve formátu PDF nebo je dostanou zástupci jednotlivých oddílů po skončení závodů od organizátora v tištěné podobě.

Nejdůležitějšími údaji ve výsledcích týkajících se samotných závodníků a jejich výsledku v určité disciplíně jsou:

- jméno,
- příjmení,
- ročník narození,
- kategorie (pohlaví)
- dosažený čas (mohou být uváděny i mezičasy),
- číslo disciplíny a její název,

- pořadí závodníků,
- číslo rozplavby a dráhy, ve které závodník plaval,
- počet dosažených bodů za dosažený čas,
- zkratka oddílu, který plavec reprezentuje.

Dále se uvádí informace o pořadateli, místě a datu konání závodů, popis bazénu (délka, počet drah) a způsob měření času (elektronická časomíra, ruční stopky). Mohou být uváděny jména rozhodčích a jejich funkce, časový rozpis jednotlivých disciplín, vysvětleny zkratky jednotlivých oddílů a další informace o závodech, které však nejsou z hlediska výsledku závodníka tolik důležité.

V oddíle TJ Tesla Brno se v současné době ukládají výsledky členů oddílu v papírové podobě do archivu nebo se neukládají vůbec a pouze se spoléhá na výsledky umístěné na webových stránkách plaveckého svazu.

Pro potřeby oddílu – ukládání výsledků svých členů - jsou některé informace zbytečné a vedení oddílu postačuje ukládat výsledný čas a umístění, kterého konkrétní plavec či štafeta plavců dosáhla.

1.15 Trenéři

Zaměstnanci oddílu jsou přiřazeni na začátku roku k jednotlivým skupinám. Většinou dvojice trenérů vede jednu skupinu. Jsou ale možné změny v průběhu sezóny, pokud se jeden či dokonce oba trenéři nemohou na trénink dostavit. V tomto případě se dopředu zajistí náhradní osoba.

O trenérech potřebuje vedení oddílu uchovávat osobní a kontaktní údaje a navíc údaje o bankovním spojení, pro zasílání výplaty. Dále výši hodinové odměny za odvedenou práci, údaje o absolvovaných kurzech a certifikátech souvisejících s činností trénování plavání. Především jde o dosaženou třídu titulu trenér plavání, rozhodčí plavání, záchrannářský kurz a kurzy zaměřené pro práci s dětmi a mládeží.

1.16 Docházka

Jeden či více z trenérů provede na začátku nebo v průběhu tréninku zápis přítomnosti či nepřítomnosti členů oddílu do připraveného listu. Také provede zápis přítomnosti

trenérů včetně té své. Součástí tabulky s docházkou je prostor pro poznámky, kam se zapisují případné změny v účasti trenérů či jiné informace. Zápis docházky se provádí formou domluvených symbolů. Možnosti jsou účast (symbol |), neúčast (symbol -) a omluvená neúčast (symbol O).

1.17 Kniha zranění a ošetřování

V současné době oddíl nevede záznamy o zranění, která se stanou při výuce. Vzhledem k nemalé pravděpodobnosti úrazu v prostorách bazénu si myslím, že by si oddíl měl záznamy o zranění a případném ošetření vést. To zejména pro případné řešení situace s pojišťovnou.

1.18 Tréninkový deník

Trenéři, kteří mají na starosti závodní plavání, zapisují do papírového sešitu či deníku náplň tréninku. Činí tak zejména proto, aby měli záznamy o vzdálenosti a jakým způsobem se v určitý den plavalo a na co se trénink zaměřoval. Nevýhodou papírové formy deníku je podrobnější analýza. Například zjištění naplavaných kilometrů pro konkrétního plavce na základě jeho docházky by bylo velmi pracné.

1.19 Plavecký trénink

Plavecký trénink se obecně skládá z několika částí. Začíná tzv. rozplaváním – část tréninku pro zahřátí a rozhýbání těla. Následuje hlavní část tréninku, která už je systematicky plánována a vymyšlená trenérem. Mohou se zde objevit měřené testy nebo plavecké cvičení na zlepšení techniky, rychlosti či vytrvalosti. Poslední částí je vyplavání, které slouží k odpočinku a uvolnění těla. Tréninková náplň se zapisuje pomocí zkratk. Zápis pak vypadá například takto:

100m rozp., 4x50m VZ, 100m vypl., 6x50m VZ N, 50m vypl., 100m VZ na čas, vypl.

1.20 Další aktivity

Oddíl plavání organizuje pro zájemce i další sportovní aktivity. V současné době je možné pro členy TJ Tesla navštěvovat lanové centrum. Tato aktivita je určena nejen pro členy plaveckého oddílu, ale i pro ostatní členy TJ Tesla. Další chystanou akcí jsou příměstské a mimoměstské tábory. Uvažuje se i o pronájmu tělocvičny a dalších aktivitách.

1.21 Shrnutí požadavků

Organizace potřebuje systém, který jí umožní především uchovávat údaje o svých zaměstnancích a členech oddílu. Dalšími informacemi, které potřebuje oddíl uchovávat pro pozdější analýzy je docházka členů na tréninky, případně náplň tréninků. Jedním z požadavků je i možnost ukládat výsledky závodníků z jednotlivých závodů. Při výuce se rovněž používají různé pomůcky, u nichž je nutné vést evidenci. Oddíl se chystá nabízet členům i jiné aktivity než je plavecká výuka, proto by bylo vhodné připravit databázi i pro tuto činnost. Posledním z požadavků je vytvořit strukturu pro zaznamenávání zranění, která se stanou při výuce.

3 Vlastní návrhy řešení

3.1 Konceptuální návrh

V této části si kladu za cíl vytvoření ER modelu. Ten se skládá z ER diagramu a datového slovníku. Abych toho byl schopen, musím nejprve identifikovat entity, vztahy mezi nimi a určit multiplicitu těchto vztahů. V dalším kroku identifikuji atributy jednotlivých entit.

3.1.1 Identifikace entit

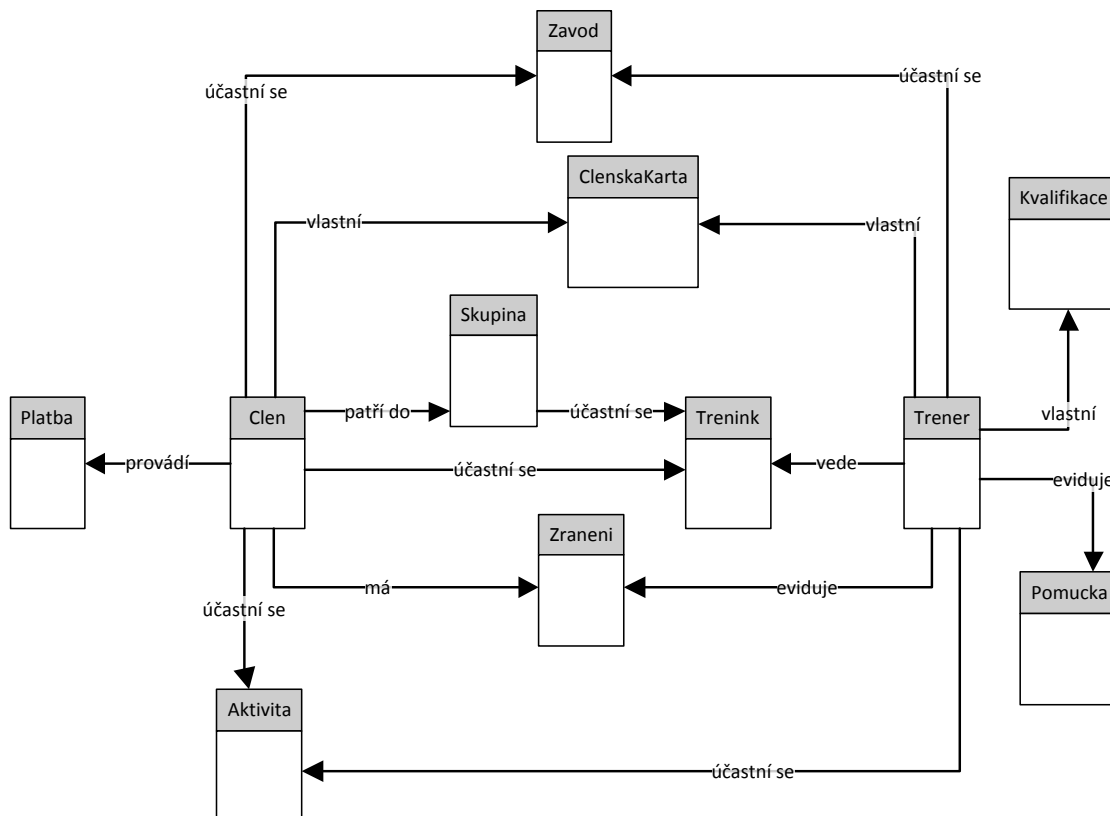
Entity zaznamenáme do tabulky, dále uvedeme jejich popis a přibližný počet výskytů. Pravděpodobně se nám nepodaří identifikovat entity všechny. To je způsobeno tím, že v konceptuální fázi ještě nedochází k dekompozici vztahů, normalizaci tabulek a dalším úpravám návrhu.

Tabulka 2: Seznam entit (Zdroj: Vlastní zpracování)

Název entity	Alias	Popis	Počet výskytů
Trener	Zaměstnanec	Zaměstnanec, osoba vedoucí výuku	10
Člen	Plavec, Závodník	Člen TJ účastníci se plavecké výuky či aktivit	150
Trenink	Výuka, výuková hodina	Jednotlivé tréninkové hodiny plavání	300 za rok
Skupina		Skupina, do níž jsou plavci přiřazeni	15
Pomucka	Majetek	Plavecké pomůcky používané při výuce, evidence jejich poškození, nákupu	100
ClenskaKarta		Karta sloužící ke vstupu do bazénu	150 za rok
Zavod		Plavecké závody, kterých se členové oddílu zúčastňují	20 za rok
Aktivita		Jiné aktivity oddílu mimo plaveckou výuku	20 za rok
Platba		Platby členů za výuku či ostatní aktivity	300 za rok
Kvalifikace	Osvědčení, certifikace	Absolvované kurzy, kvalifikace, certifikace související s činností trénování	5
Zraneni	Úraz, incident	Zranění, ke kterým dojde při výuce	20 za rok

3.1.2 Identifikace vztahů mezi entitami

V tomto kroku se snažíme najít všechny vztahy mezi entitami identifikovanými v předchozím kroku. Zároveň určíme multiplicitu vztahů mezi entitami. Následně vytvoříme první verzi ER diagramu.



Obrázek 2: ER diagram - první verze (Zdroj: vlastní zpracování)

3.1.3 Identifikace atributů, určení klíčů

V této fázi nejprve identifikujeme atributy veškerých nalezených entit. Tyto atributy zaznamenáme do slovníku dat včetně jejich popisu, typu a délky. Do slovníku též zaznačíme pro všechny atributy povolení nabývat hodnotu NULL. Dalšími kroky bude určení domén atributů, kandidátních, primárních a alternativních klíčů.

Při identifikaci atributů jsem dospěl ke zjištění, že z některých atributů bude vhodné vytvořit entity. Z tohoto důvodu jsem musel předchozí část konceptuálního návrhu opakovat a vytvořit nový ER diagram.

Tabulka 3: Vztahy mezi entitami (Zdroj: vlastní zpracování)

Entita	Multiplicita	Vztah	Multiplicita	Entita
Clen	1..1	Vlastní	0..*	ClenskaKarta
Clen	1..1	provádí	0..*	Platba
Clen	0..*	patří do	0..*	TreninkovaSkupina
Clen	0..*	účastní se	0..*	Trenink
Clen	0..*	účastní se	0..*	Aktivita
Clen	1..*	účastní se	0..*	Zavod
Clen	1..4	dosahuje	0..*	Vysledek
Clen	1..1	je	0..1	Trener
Clen	2..2	má	0..*	ZaznamOZraneni
Trener	1..1	vytváří	0..*	ZaznamOPoskozeni
Trener	1..*	vlastní	0..*	Kvalifikace
Dodavatel	1..1	dodává	1..*	Pomucka
Pomucka	1..1	má	0..*	ZaznamOPoskozeni
Zavod	1..1	má	0..*	Vysledek
TreninkovaSkupina	1..1	má	0..*	Trenink
Trenink	1..1	skládá se z	0..*	CastTreninku

3.1.4 Clen

Entita **Clen** reprezentuje všechny členy oddílu plavání včetně trenérů a těch členů, kteří se zúčastňují pouze jiných aktivit než výuky plavání. Obsahuje především osobní a kontaktní údaje o jednotlivých členech oddílu. Doménou atributů **Jmeno** a **Prijmeni** jsou znakové řetězce s proměnlivou délkou o maximální délce 15 znaků pro jméno a 20 znaků pro příjmení osoby. Pro atribut **RodCislo** jsou přípustnými znaky číslice a znak lomítka, délka tohoto řetězce může být 10 či 11 znaků. Atribut **Ulice** je tvořen znakovým řetězcem o maximální délce 40 znaků. **CP** je řetězec znaků o maximální délce 6 znaků a obsahuje popisné číslo domu, kde člen bydlí. **Mesto** je znakový řetězec o maximální délce 40 znaků. **PSC** je znakový řetězec o délce 5 znaků a obsahuje

poštovní směrovací číslo. **Telefon** je znakový řetězec o délce maximálně 14 znaků. Může obsahovat pouze číselné hodnoty a znak plus. Lze tedy uložit telefonní číslo v mezinárodním formátu. Atribut **Email** slouží k uložení e-mailové adresy. Doménou tohoto atributu je znakový řetězec a délka tohoto řetězce je maximálně 60 znaků. Každý člen může uvést pouze jeden kontaktní telefon a jednu e-mailovou adresu. **DatNar** je určen k uložení data narození člena. **Pohlavi** uchovává informaci o pohlaví osoby. Může nabývat jedné z logických hodnot (např. 1 pro muže, 0 pro ženu; pravda pro muže, nepravda pro ženu). **SUdaje**, **SOddil**, **SInfo** a **SFoto** jsou atributy mající jednu z logických hodnot a slouží k zjištění, zda člen oddílu souhlasí se zpracováním osobních údajů, podmínkami oddílu, zasíláním informací o činnosti oddílu a s uveřejňováním fotografií, na kterých je člen zachycen, na webové stránky oddílu. Dalšími atributy jsou **DatRegOdd**, který znamená datum registrace člena do oddílu a **DatRegSvaz**, který značí datum registrace plavce na plavecký svaz. Oba atributy jsou typu datum. Posledním atributem je **Plavec**, jehož logická hodnota nás informuje o tom, zda se člen oddílu zároveň účastní plavecké výuky (logická hodnota – pravda) nebo jen jiných aktivit (logická hodnota – nepravda).

Kandidátním klíčem této entity je složený klíč skládající se z atributů **Jmeno+Prijmeni+DatNar+Telefon**. Kde datem narození vylučujeme možnost, kdy by například otec a syn měli stejné křestní jméno. Telefonním číslem bychom vyloučili možnost, že by existovaly dvě osoby se stejným jménem a příjmením narozené ve stejný den. Místo atributu **Telefon** by se daly použít atributy **Ulice** a **CP**. Dalším kandidátním klíčem je **RodCislo**. To by mělo k jednoznačnému určení osoby stačit. Nicméně jsem se rozhodl vytvořit pro každého člena umělý klíč **IDClen**, který bude jednoznačně identifikovat každého člena. Doménou tohoto atributu je číslo, které je pro každého nového člena automaticky generováno. **IDClen** tedy bude primárním klíčem. Ostatní kandidátní klíče budou klíči alternativními.

Clen		
PK	IDClen	N-Automatické počítadlo(1)
	Jmeno	C-Proměnlivá délka(15)
	Prijmeni	C-Proměnlivá délka(20)
	RodCislo	C-Pevná délka(11)
	Ulice	C-Proměnlivá délka(40)
	CP	C-Proměnlivá délka(6)
	Mesto	C-Proměnlivá délka(40)
	PSC	C-Pevná délka(5)
	Telefon	C-Proměnlivá délka(14)
	Email	C-Proměnlivá délka(60)
	DatNar	T-Datum
	Pohlavi	L-True nebo False
	SUdaje	L-True nebo False
	SOddil	L-True nebo False
	SInfo	L-True nebo False
	SFoto	L-True nebo False
	DatRegOdd	T-Datum
	DatRegSvaz	T-Datum
	Plavec	L-True nebo False

Obrázek 3: Entita Clen (Zdroj: vlastní zpracování)

3.1.5 Trener

Tato entita specializuje entitu **Clen**. Trenér je členem oddílu, ale ne všichni členové oddílu jsou zároveň trenéři. O trenérech potřebuje oddíl kromě těch údajů, které již obsahuje entita **Clen**, uchovávat číslo bankovního účtu, výši hodinové odměny a informace o absolvovaných kurzech a vlastněných certifikátech. **IDTrener** je osobní číslo zaměstnance a pro stejnou osobu je shodné s číslem **IDClen** entity **Clen**. **CisloUctu** je znakový řetězec o pevné délce 22 znaků, který obsahuje číslo účtu v plném formátu. Ten se skládá z 6 číslic na začátku následovaných pomlčkou, 10 číslicemi, lomítkem a 4 číslicemi na jeho konci. **Odmena** je atribut, který obsahuje nezáporné desetinné číslo, udávající výši hodinové odměny zaměstnance.

Kandidátním klíčem jsou atributy **CisloUctu**, protože každý zaměstnanec musí mít vlastní bankovní účet. Dalším je **IDTrener**, který jednoznačně identifikuje každého zaměstnance oddílu. Z těchto kandidátních klíčů jsem vybral jako primární klíč **IDTrener**

Trener		
PK	IDTrener	N-Celé číslo bez znaménka
	Jmeno	C-Proměnlivá délka(15)
	Prijmeni	C-Proměnlivá délka(20)
	CisloUctu	C-Pevná délka(22)
	Odmena	N-Peníze(6,2)

Obrázek 4: Entita Trener (Zdroj: vlastní zpracování)

3.1.6 Kvalifikace

Kvalifikace slouží k uchování informací o dosažené kvalifikaci trenéra a certifikátech, které trenér vlastní. Skládá se z atributů **IDKvalifikace** a **Nazev**. **IDKvalifikace** je jednoznačná identifikace kvalifikace pomocí její zkratky. Je to řetězec s proměnlivou délkou o maximálně pěti znacích. Tento atribut jsem zvolil primárním klíčem. Atribut **Nazev** je znakovým řetězcem o maximální délce 100 znaků a obsahuje plný název certifikace.

Kvalifikace		
PK	<u>IDKvalifikace</u>	C-Proměnlivá délka(5)
	Nazev	C-Proměnlivá délka(100)

Obrázek 5: Entita Kvalifikace (Zdroj: vlastní zpracování)

3.1.7 Platba

Entita **Platba** popisuje platby, které provádí člen oddílu. Tyto platby jsou většinou příspěvky za výukové hodiny nebo za aktivity, kterých se člen účastní. Mohou se zde ale objevit i platby za různé vybavení, které si člen může od oddílu zakoupit (například oddílové oblečení). Atribut **Ucel** specifikuje účel platby. Jde o znakový řetězec o délce až 255 znaků. **Castka** umožňuje zaznamenání částky k zaplacení. Jde o číselnou hodnotu s až 2 desetinnými místy a celkové délce až 8 znaků. **DatumUhr** určuje datum zaplacení celé částky. Primárním klíčem je uměle vytvořený klíč - atribut **IDPlatba**, který obsahuje nezáporné číslo jednoznačně určující každou platbu. Atribut **VarSymbol** má pevnou délku 8 znaků a skládá se ze znakového řetězce, kde první dva znaky znamenají kód oddílu (16 – kód oddílu plavání), další 4 znaky pozice nabývají číselné hodnoty od 0000 do 9999 a obsahují osobní číslo člena. Dva následující znaky určují typ platby (např. 00 – platba příspěvků za plavání, 01 – platba za lanové centrum atd.)

Platba		
PK	<u>IDPlatba</u>	N-Automatické počítadlo(1)
	VarSymbol Ucel Castka DatumUhr	C-Pevná délka(8) C-Proměnlivá délka(255) N-Peníze(8,2) T-Datum

Obrázek 6: Entita Platba (Zdroj: vlastní zpracování)

3.1.8 ClenskaKarta

Členská karta obsahuje jméno (**Jmeno**) a příjmení (**Prijmeni**) vlastníka karty. Oba atributy jsou znakové řetězce o proměnlivé délce. Dále datum platnosti (**Platnost**) a **IDKarta** – číselná hodnota jednoznačně identifikující každý exemplář členské karty. Primárním klíčem entity **ClenskaKarta** je právě **IDKarta**.

ClenskaKarta		
PK	<u>IDKarta</u>	N-Celé číslo bez znaménka
	Jmeno Prijmeni Platnost	C-Proměnlivá délka(15) C-Proměnlivá délka(20) T-Datum

Obrázek 7: Entita ClenskaKarta (Zdroj: vlastní zpracování)

3.1.9 TreninkovaSkupina

Popisuje každou tréninkovou skupinu. **IDSkupina** je jednoznačné určení každé tréninkové skupiny. Kód každé skupiny se skládá z dvou číslic pro určení roku (např. 13 pro rok 2013), jedné číslice pro určení pololetí (1 – první, 2 – druhé), jedné číslice pro určení dne, kdy probíhá výuka této skupiny (1 – pondělí, 2 – úterý atd.), jednoho znaku, který specifikuje skupinu (A – závodní družstvo, B – závodní příprava, N – neplavci, Z – začátečníci, P – pokročilí, D – dospělí) a poslední je pořadové číslo (0-9), pokud ve stejný den trénuje více stejných skupin. Kód 1312N0 tedy značí skupinu neplavců, kteří mají trénink v úterý, v prvním pololetí roku 2013 a jedná se ten den o první skupinu neplavců. **Rok** značí rok, ve kterém probíhá výuka skupiny. Jde o číselnou hodnotu. **Pololetí** značí první respektive druhou polovinu roku. Zde je pololetí reprezentováno logickou hodnotou (0 – první pololetí a 1 – druhé pololetí). **Den** je atribut označující den v týdnu, kdy má skupina výukovou hodinu. Jde o znakový řetězec o pevné délce 2 znaky (Po – pondělí, Út – úterý atd.). **Nazev** specifikuje danou skupinu. Jde o znakový řetězec proměnlivé délky o maximálním rozsahu 30 znaků. **Popis** detailně popisuje danou tréninkovou skupinu. Jedná se o dlouhý řetězec znaků o proměnlivé délce. **CasZac** a **CasKon** jsou atributy označující začátek resp. konec výukové hodiny. Datovým typem je čas ve tvaru HH:MM (H – hodina, M – minuta). Kandidátním klíčem je složený klíč **Rok+Pololetí+Den+Nazev+CasZac** a jednoduchý klíč **IDSkupina**. Jako primární klíč jsem zvolil **IDSkupina** a složený klíč se tedy stal klíčem alternativním.

TreninkovaSkupina		
PK	<u>IDSkupina</u>	C-Pevná délka(6)
	Rok Pololetí Den Název Popis CasZac CasKon	C-Pevná délka(4) L-True nebo False C-Pevná délka(2) C-Proměnlivá délka(30) C-Rozsáhlá délka T-Čas T-Čas

Obrázek 8: Entita TreninkovaSkupina (Zdroj: vlastní zpracování)

3.1.10 Trenink

Entita **Trenink** představuje jednotlivé výukové hodiny plavání. Každá tréninková hodina je jednoznačně určena složeným klíčem **IDSkupina+Datum**

Trenink		
PK	<u>Datum</u>	T-Datum
PK	<u>IDSkupina</u>	C-Pevná délka(6)

Obrázek 9: Entita Trenink (Zdroj: vlastní zpracování)

3.1.11 CastTreninku

Tato entita reprezentuje každý prvek plaveckého tréninku (část tréninkové náplně). Každá část tréninku je jednoznačně určena složeným klíčem **IDSkupina+Datum+Poradi**. Atribut **Poradi** určuje pořadí každé části, ze které se tréninková náplň skládá. Nabývá kladných číselných hodnot. Pořadí částí je důležité pro pozdější zobrazení dat. Neuspořádaný zápis tréninku by mohl mít jiný význam, než byl ten původní. **Opakovani** značí počet opakování při plavání stejné délky, stejným způsobem. Obsahuje pouze číselnou hodnotu a také pouze kladné číslo. Atribut **Delka** slouží k zaznamenání uplavané délky v metrech. Povolená jsou pouze celá kladná čísla. **Zpusob** značí plavecký způsob, kterým se daná délka plavecké části plave. Atribut se skládá ze znakového řetězce s pevnou délkou 2 znaky. Hodnotami atributu jsou zkratky názvů plaveckých způsobů (VZ – volný způsob, P – prsa, Z – znak, M – motýlek, PZ – polohový závod). **NohyRuce** je atribut mající význam nás informovat o tom, jestli se část plavala pouze nohama či rukami. Atribut může mít hodnotu znaku N (nohy), R (ruce) či NULL. Atribut **TC** nabývá logické hodnoty pravda (1), pokud se část tréninku plavala tzv. technickým cvičením a logické hodnoty nepravda (0) v opačném případě. Stejně pravidlo platí pro atributy **Ploutve** a **Packy**, které označují plavání s pomůckami

na nohou respektive rukou plavce. **RozpVypl** je znakový řetězec pevné délky 4 znaky, informující o tom, jestli daná část je rozplaváním (část na začátku tréninku sloužící k rozhybání těla) či vyplaváním (část na konci nebo i v průběhu tréninku, sloužící k uvolnění po výkonu). Tento atribut nabývá hodnoty rozp – rozplavání, vypl – vyplavání či NULL pokud nejde ani o jednu z předchozích variant. Posledním atributem této entity je **Poznamka**, sloužící k upřesňujícímu popisu plavané části. Může obsahovat například údaje o tempu, v jakém se daná část plavala. Tento atribut je znakovým řetězcem s proměnlivou délkou.

CastTreninku		
PK	<u>Datum</u>	T-Datum
PK	<u>IDSkupina</u>	C-Pevná délka(6)
PK	<u>Poradi</u>	N-Celé číslo bez znaménka
	Opakovani	N-Celé číslo bez znaménka
	Delka	N-Celé číslo bez znaménka
	Zpusob	C-Pevná délka(2)
	NohyRuce	C-Pevná délka(1)
	TC	L-True nebo False
	Ploutve	L-True nebo False
	Packy	L-True nebo False
	RozpVypl	C-Pevná délka(4)
	Poznamka	C-Proměnlivá délka(50)

Obrázek 10: Entita CastTreninku (Zdroj: vlastní zpracování)

3.1.12 Aktivita

Tato entita popisuje aktivity pořádané plaveckým oddílem jako například návštěvy lanového centra. Primárním klíčem je uměle vytvořený klíč **IDAktivita**, který obsahuje automaticky generované číslo. Atribut **Kod** obsahuje kódové označení určité aktivity (například LC pro lanové centrum). **Kod** je znakový řetězec proměnlivé délky o maximálně pěti znacích. **Nazev** obsahuje název dané aktivity (například Lanové centrum). Tento atribut může obsahovat až 50 znaků. **Popis** detailně popisuje aktivitu. Jedná se o nepovinnou položku, která může nabývat velmi dlouhý znakový řetězec. **Zacatek** a **Konec** slouží k uložení data a času začátku a konce dané aktivity.

Aktivita		
PK	<u>IDAktivita</u>	N-Automatické počítadlo(1)
	Kod	C-Proměnlivá délka(5)
	Nazev	C-Proměnlivá délka(50)
	Popis	C-Rozsáhlá délka
	Zacatek	T-Datum a čas
	Konec	T-Datum a čas

Obrázek 11: Entita Aktivita (Zdroj: vlastní zpracování)

3.1.13 ZaznamOZraneni

ZaznamOZraneni je entita obsahující atributy nezbytné pro zaznamenání každého úrazu, který se stane při výuce plavání či aktivitách. Primárním klíčem je automaticky generované číslo atributu **IDZaznam**. **DatumCas** slouží k zápisu data a času, kdy došlo k incidentu. **Popis** poskytuje detailní popis incidentu včetně informací o ošetření. Tento atribut umožňuje uložit velmi dlouhý řetězec znaků. Dalšími atributy jsou **JmenoZraneny** a **JmenoTrener** obsahující celá jména zraněného a trenéra, který incident řešil. Alternativním klíčem je složený klíč **DatumCas+JmenoZraneny**.

ZaznamOZraneni		
PK	IDZaznam	N-Automatické počítadlo(1)
	DatumCas Popis JmenoZraneny JmenoTrener	T-Datum a čas C-Rozsáhlá délka C-Proměnlivá délka(35) C-Proměnlivá délka(35)

Obrázek 12: Entita ZaznamOZraneni (Zdroj: vlastní zpracování)

3.1.14 Zavod

Tato entita reprezentuje každý závod, kterého se plavci oddílu zúčastní. **Nazev** obsahuje celé jméno konkrétního závodu (například Krajský přebor dorostu a dospělých nebo Memoriál Jaroslava Jezbery – 26. ročník). **Mesto** a **Bazen** specifikují místo, kde se závod koná, přičemž druhý jmenovaný atribut není nutné zadávat, ale je zde pro případ rozlišení měst, ve kterých je bazénů více. Všechny tyto atributy jsou typu znakový řetězec s proměnlivou délkou. **DatumOd** a **DatumDo** jsou atributy informující o datu začátku závodu a jeho konci (pokud je vícedenní). **DelkaBazenu** je atribut určující délku bazénu v metrech. Běžnými hodnotami jsou 25 nebo 50m. Datovým typem je číslo malého rozsahu. **TypMereni** je atribut, jehož účelem je sdělovat, jakým způsobem je závod měřen. Závody jsou měřeny ručními stopkami, nebo elektronickou časomírou. Primárním klíčem této entity je uměle vytvořený klíč, který obsahuje jedinečnou číselnou hodnotu.

Zavod		
PK	IDZavod	N-Automatické počítadlo(1)
	Nazev Mesto Bazen DatumOd DatumDo DelkaBazenu TypMereni	C-Proměnlivá délka(100) C-Proměnlivá délka(40) C-Proměnlivá délka(40) T-Datum T-Datum N-Celé číslo bez znaménka C-Proměnlivá délka(30)

Obrázek 13: Entita Zavod (Zdroj: vlastní zpracování)

3.1.15 Vysledek

Vysledek je entita, která slouží k uchování výkonů dosažených při závodech konkrétními členy oddílu. Atribut **Disciplina** obsahuje jednu z plaveckých disciplín (například 200m volný způsob). Dalšími atributy jsou **Jmeno1** a **Prijmeni1** obsahující znakové řetězce - jméno a příjmení plavce, který daný výkon zaplavával. V případě štafetového závodu se jedná o jméno a příjmení prvního závodníka. **CasCil** slouží k zaznamenání výsledného času závodníka či štafety. Doménou atributu jsou časové údaje obsahující kromě hodin, minut a sekund i setiny sekund. **Umisteni** je atribut pro zaznamenání dosaženého umístění závodníka v konečném pořadí. **Jmeno2**, **Prijmeni2**, **Jmeno3**, **Prijmeni3**, **Jmeno4**, **Prijmeni4** jsou jména a příjmení zbývajících členů štafety. **Cas1** je čas dohmatu prvního člena štafety, který lze považovat, na rozdíl od časů ostatních členů štafety, za platný výkon, který lze použít při přihlašování na závody, na něž je potřeba se kvalifikovat. **Datum** je určen pro zaznamenání data, kdy byl výkon zaplavan. Primárním klíčem je atribut **IDVysledek**, který je uměle vytvořeným klíčem, obsahující automaticky generované číselné hodnoty.

Vysledek		
PK	IDVysledek	N-Automatické počítadlo(1)
	Disciplina Jmeno1 Prijmeni1 CasCil Umisteni Jmeno2 Prijmeni2 Jmeno3 Prijmeni3 Jmeno4 Prijmeni4 Cas1 Datum	C-Proměnlivá délka(50) C-Proměnlivá délka(15) C-Proměnlivá délka(20) T-Čas N-Celé číslo bez znaménka C-Proměnlivá délka(15) C-Proměnlivá délka(20) C-Proměnlivá délka(15) C-Proměnlivá délka(10) C-Proměnlivá délka(15) C-Proměnlivá délka(20) T-Čas T-Datum

Obrázek 14: Entita Vysledek (Zdroj: vlastní zpracování)

3.1.16 Pomucka

Pomucka je entita reprezentující pomůcky a jiný materiál používaný při výuce. Každá pomůcka má přidělený jednoznačný kód, který se skládá ze dvou znaků - písmen na začátku řetězce a čtyř číslic (například DV0001 – kódové označení pro velkou desku s pořadovým číslem 0001). Tento kód – atribut **IDPomucka** je zároveň primárním klíčem entity. Dále má každá pomůcka název, může mít krátký popis a specifikovanou barvu. U větších pomůcek se vytváří záznamy pro každou pomůcku zvlášť, ale pro malé pomůcky například tzv. vodní samolepky vytvoříme pro všechny pomůcky stejného typu a stejného data nákupu jeden záznam a zapíšeme údaj o počtu kusů. K tomu slouží atribut **Pocet**. U větších pomůcek zadáme počet kusů roven jedné. Dále je možné u pomůcek zaznamenávat datum pořízení (atribut **DatumNakupu**) a cenu za jednotku (atribut **Cena**).

Pomucka		
PK	<u>IDPomucka</u>	C-Pevná délka(6)
	Nazev Popis Barva Pocet DatumNakup Cena	C-Proměnlivá délka(20) C-Proměnlivá délka(255) C-Proměnlivá délka(10) N-Celé číslo bez znaménka T-Datum N-Peníze(8,2)

Obrázek 15: Entita Pomucka (Zdroj: vlastní zpracování)

3.1.17 Dodavatel

O dodavateli pomůcek stačí oddíl uchovávat, mimo jeho jména (**Nazev**), kontaktní informace, jakými jsou telefonní číslo (**Telefon**), e-mailová adresa (**Email**) a adresa webových stránek (**Web**). Poštovní adresu není potřeba uchovávat, protože objednávání pomůcek většinou probíhá telefonicky nebo prostřednictvím e-mailové korespondence. Každému dodavateli je přiděleno osobní číslo, které ho jednoznačně identifikuje. Osobní číslo je uloženo do atributu **IDDodavatel**, který je primárním klíčem entity.

Dodavatel		
PK	<u>IDDodavatel</u>	N-Automatické počítadlo(1)
	Nazev Telefon Email Web	C-Proměnlivá délka(30) C-Proměnlivá délka(14) C-Proměnlivá délka(60) C-Proměnlivá délka(50)

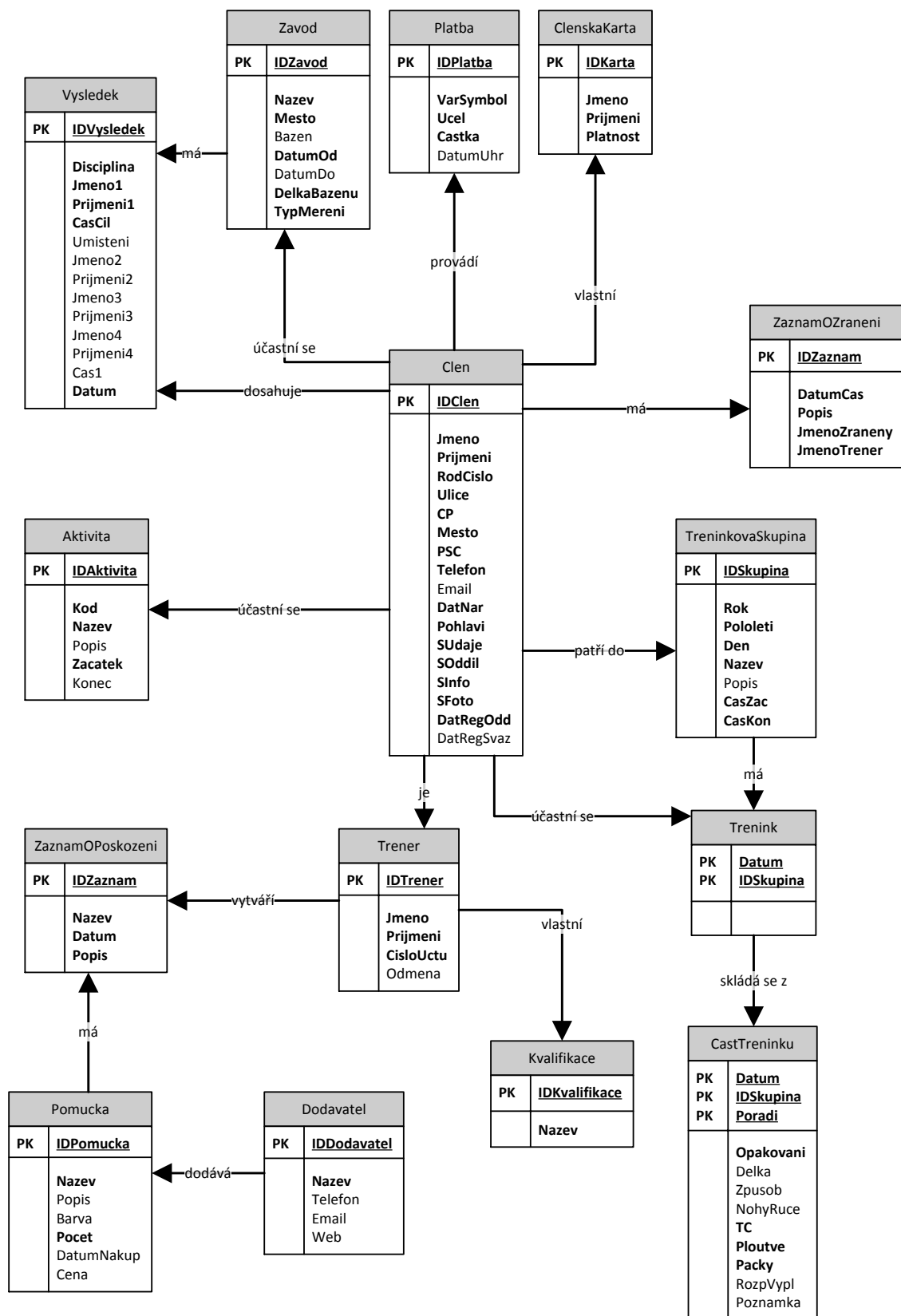
Obrázek 16: Entita Dodavatel (Zdroj: vlastní zpracování)

3.1.18 ZaznamOPoskozeni

Záznamy o poškození se vyplňují v případě zjištění poškození pomůcky. Záleží na vedení oddílu, zdali bude pomůcky kontrolovat po každé výukové hodině nebo například jednou týdně. V každém záznamu musí být uveden název pomůcky (**Nazev**), datum vytvoření záznamu (**Datum**) a slovní popis (**Popis**) detailně informující o rozsahu poškození. Primárním klíčem je atribut **IDZaznam** - umělý klíč obsahující automaticky generované číslo.

ZaznamOPoskozeni		
PK	<u>IDZaznam</u>	N-Automatické počítadlo(1)
	Nazev Datum Popis	C-Proměnlivá délka(30) T-Datum C-Rozsáhlá délka

Obrázek 17: Entita ZaznamOPoskozeni (Zdroj: vlastní zpracování)



Obrázek 18: ER diagram - konceptuální návrh (Zdroj: vlastní zpracování)

3.2 Logický návrh

V této části návrhu databáze vycházím z konceptuálního návrhu – ER diagramu. Cílem logického návrhu každou entitu reprezentovat pomocí tabulky, zajistit spojení pomocí cizích klíčů a provést normalizaci tabulek.

3.2.1 ClenskaKarta

ClenskaKarta je ve vztahu s entitou Clen. Platí pro ni, že jeden člen oddílu může vlastnit více členských karet a každá členská karta přísluší pouze jednomu konkrétnímu členu. V tomto případě je Clen rodičovská entita a ClenskaKarta je dceřiná, proto vytvoříme v tabulce ClenskaKarta kopii primárního klíče entity Clen.

Tabulka je v první normální formě (1NF), protože neobsahuje vícehodnotové atributy. Vzhledem k tomu, že druhá normální forma se týká tabulek se složeným primárním klíčem a tato tabulka obsahuje jednoduchý primární klíč, můžeme prohlásit, že je tabulka i v druhé normální formě (2NF). Tabulka však porušuje třetí normální formu (3NF), protože atributy Jmeno a Prijmeni jsou závislé na cizím klíči IDClen, proto provedeme jejich odstranění.

ClenskaKarta	IDKarta, Platnost, IDClen
Primární klíč	IDKarta
Cizí klíč	IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání kaskáda

3.2.2 Platba

Vztah mezi entitami Platba a Clen je definován tak, že jeden člen provádí více plateb, ale jednu platbu provádí jeden člen. Opět zvolíme cizím klíčem kopii primárního klíče entity Clen, kterou umístíme do tabulky Platba.

Variabilní symbol je složený atribut a proto ho rozdělíme na 3 nové atributy – VSOdd obsahující kód oddílu, VSClen obsahující osobní číslo člena, VSKod obsahující kód účelu platby. Tím splníme podmínky první normální formy. Tabulka je v 2NF, jelikož neobsahuje složený primární klíč. Třetí normální formy dosáhneme tak, že Atribut VSClen vypustíme, neboť nám stejnou informaci dává atribut IDClen. Dále vytvoříme novou tabulku Ucel, která bude obsahovat atributy VSKod a Ucel. Spojení zajistíme cizím klíčem v tabulce Platba.

Platba	IDPlatba, VSOdd, VSKod, Castka, DatumUhr, IDClen
Primární klíč	IDPlatba
Cizí klíč	IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce, VSKod reference Ucel (VSKod) při aktualizaci kaskáda, při vymazání žádná akce

Ucel	VSKod, Ucel
Primární klíč	VSKod

3.2.3 Trener

Každý trenér je zároveň členem oddílu, ale ne každý člen je trenérem. Proto umístíme kopii primárního klíče entity Clen do tabulky Trener. Tabulka je v 1NF i 2NF. Třetí normální formu porušují atributy Jmeno a Prijmeni. Ty jsou tranzitivně závislé na atributu IDClen. Proto tyto atributy z tabulky odstraníme a tím dosáhneme 3NF.

Trener	IDTrener, CisloUctu, Odmena, IDClen
Primární klíč	IDTrener
Alternativní klíč	CisloUctu
Cizí klíč	IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

3.2.4 Kvalifikace

Entita kvalifikace je ve vzájemném vztahu s entitou Trener. Platí, že určitou kvalifikaci může vlastnit více trenérů a také, že jeden trenér může vlastnit více kvalifikací. Proto provedeme dekompozici a vytvoříme pomocnou tabulku, do které umístíme kopie primárních klíčů obou entit. Tabulku jsem nazval TrenerskaKvalifikace. Dále jsem do tabulky umístil nový atribut Platnost, který uchovává informaci o datu vypršení platnosti dané kvalifikace pro konkrétního člena. Datovým typem je jednoduché datum.

Obě tabulky jsou v první, druhé i třetí normální formě, tudíž normalizaci provádět nemusíme.

Kvalifikace	IDKvalifikace, Nazev
Primární klíč	Nazev

TrenerskaKvalifikace	IDTrener, IDKvalifikace, Platnost
Primární klíč	IDTrener, IDKvalifikace
Cizí klíč	IDTrener reference Trener (IDTrener) při aktualizaci kaskáda, při vymazání kaskáda, IDKvalifikace reference Kvalifikace (IDKvalifikace) při aktualizaci kaskáda, při vymazání kaskáda

3.2.5 TreninkovaSkupina

Člen oddílu může patřit do více tréninkových skupin a každá tréninková skupina může mít a zpravidla má více členů. Opět provádíme dekompozici vztahu a do nově vzniklé tabulky vložíme kopie primárních klíčů obou entit. Novou tabulku jsem nazval Prislusnost.

Obě tabulky neporušují ani jednu z prvních tří normálních forem, proto normalizaci neprovádíme.

TreninkovaSkupina	IDSkupina, Rok, Pololeti, Den, Nazev, Popis, CasZac, CasKon
Primární klíč	IDSkupina
Alternativní klíč	Rok+Pololeti+Den+Nazev+CasZac

Prislusnost	IDSkupina, IDClen
Primární klíč	IDSkupina, IDClen
Cizí klíč	IDSkupina reference TreninkovaSkupina (IDSkupina) při aktualizaci kaskáda, při vymazání žádná akce, IDClen reference clen (IDClen) při aktualizaci kaskáda, při vymazání kaskáda

3.2.6 Trenink

Entita trenink je ve vztahu ke dvěma dalším entitám – Clen a TreninkovaSkupina. Tréninková skupina má více tréninků, jednoho tréninku se určitá skupina účastní jednou. Proto zkopírujeme primární klíč entity TreninkovaSkupina a umístíme ho do tabulky Trenink. Vztah mezi tréninkem a členem je definován tak, že jeden člen se účastní více tréninků a jednoho tréninku se může zúčastnit více členů. Proto vytvoříme novou tabulku, kterou nazveme DochazkaTrenink. Do nové tabulky zkopírujeme primární klíče entit Trenink a Clen. Do této tabulky ještě umístím atribut Ucast, který bude typu znakový řetězec pevné délky jeden znak. Tento atribut bude nabývat hodnoty U při účasti a O při omluvené neúčasti člena na tréninku.

Obě tabulky jsou ve všech třech normálních formách, proto normalizaci neprovádím.

Trenink	Datum, IDSkupina
Primární klíč	Datum, IDSkupina
Cizí klíč	IDSkupina reference TreninkovaSkupina (IDSkupina) při aktualizaci kaskáda, při vymazání žádná akce

DochazkaTrenink	Datum, IDSkupina, IDClen, Ucast
Primární klíč	Datum, IDSkupina, IDClen
Cizí klíč	Datum, IDSkupina reference Trenink (Datum, IDSkupina) při aktualizaci kaskáda, při vymazání žádná akce, IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

3.2.7 CastTreninku

Každý trénink se skládá z jedné nebo více částí, jedna část přísluší jednomu tréninku. Proto do tabulky CastTreninku musíme zkopírovat primární klíč entity Trenink.

Tabulka je v první, druhé i třetí normální formě.

CastTreninku	Datum, IDSkupina, Poradi, Opakovani, Delka, Zpusob, NohyRuce, TC, Ploutve, Packy, RozplVypl, Poznamka
Primární klíč	Datum, IDSkupina, Poradi
Cizí klíč	Datum, IDSkupina reference Trenink (Datum, IDSkupina) při aktualizaci kaskáda, při vymazání kaskáda

3.2.8 Aktivita

Pro vztah aktivity a člena oddílu platí, že jeden člen se může zúčastnit více aktivit a jedné aktivity se může zúčastnit více členů. Vytvoříme proto novou tabulku s názvem DochazkaAktivita, do které umístíme kopie primárních entit Clen a Aktivita.

Tabulka Aktivita porušuje 3NF, protože atributy Nazev a Popis jsou závislé na neklíčovém atributu Kod. Normalizaci provedeme vytvořením nové tabulky s názvem AktivitaPopis. Primárním klíčem bude atribut Kod a jeho kopie bude tvořit cizí klíč v tabulce aktivita. Dále do tabulky přemístíme atributy Nazev a Popis

Aktivita	IDAktivita, Kod, Zacatek, Konec
Primární klíč	IDAktivita
Cizí klíč	Kod reference AktivitaPopis (Kod) při aktualizaci kaskáda, při vymazání žádná akce

DochazkaAktivita	IDAktivita, IDClen
Primární klíč	IDAktivita, IDClen
Cizí klíč	IDAktivita reference Aktivita (IDAktivita) při aktualizaci kaskáda, při vymazání žádná akce, IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

AktivitaPopis	Kod, Nazev, Popis
Primární klíč	Kod

3.2.9 ZaznamOZraneni

Platí, že jeden člen se může vícekrát zranit a jeden záznam odpovídá dvěma členům – zraněnému a trenérovi, který incident řešil. Novou tabulku jsem nazval ClenZaznam a umístím do ní kopie primárních klíčů entit Clen a ZaznamOZraneni. Aby bylo poznat, který ze zaznamenaných klíčů patří zraněnému a který trenérovi, přidal jsem do tabulky ClenZaznam ještě atribut Zraneny, který nabývá logické hodnoty pravda, právě když se jedná o zraněného a logické hodnoty nepravda, pokud se jedná o trenéra.

Normalizaci tabulky ZaznamOZraneni provedeme rozložením atributů JmenoZraneny a JmenoTrenér na jednoduché atributy. Tím splníme první normální formu. Tabulka je také ve druhé normální formě. Abychom dostali tabulku do 3NF, stačí atributy vytvořené v prvním kroku normalizace odstranit, neboť se jedná o tranzitivně závislé atributy. Tabulka ClenZaznam je ve všech třech normálních formách.

ZaznamOZraneni	IDZaznam, DatumCas, Popis
Primární klíč	IDZaznam

ClenZaznam	IDZaznam, IDClen, Zraneny
Primární klíč	IDZaznam, IDClen
Cizí klíč	IDZaznam reference Zaznam (IDZaznam) při aktualizaci kaskáda, při vymazání kaskáda, IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

3.2.10 Vysledek

Entita Vysledek je ve vztahu s entitami Zavod a Clen. Platí, že jednoho výsledku bylo dosaženo na jednom závodě a každý závod může mít více výsledků. Zkopírujeme primární klíč entity Zavod a umístíme ho do tabulky Vysledek. Vztah mezi závodníkem (členem) je definován tak, že jednoho výsledku může dosáhnout až čtveřice plavců a jeden plavec může dosáhnout více výsledků. Opět nezbývá než vytvořit pomocnou tabulku, do které umístíme kopie primárních klíčů entit Clen a Vysledek. Tuto tabulku

jsem nazval VysledekClen. Umístil jsem do ní ještě atribut Poradi, který obsahuje číselnou hodnotu a ukládá informaci o pořadí člena ve štafetě.

Tabulka Vysledek obsahuje složený atribut Disciplina, který se skládá ze vzdálenosti a plaveckého způsobu. Vytvoříme proto novou tabulku Disciplina s atributy Vzdalenost a Zpusob a primárním klíčem DisciplinaZkratka obsahujícím zkrácený název disciplíny. 2NF tabulka neporušuje. Pro splnění pravidel 3NF musíme odstranit z tabulky Vysledek všechny atributy týkající se jmen závodníků. K jejich identifikaci nyní slouží záznamy v tabulce VysledekClen.

Tabulka VysledekClen je ve všech třech normálních formách.

Vysledek	IDVysledek, DisciplinaZkratka, CasCil, Umistení, Cas1, Datum, IDZavod
Primární klíč	IDVysledek
Cizí klíč	IDZavod reference Zavod (IDZavod) při aktualizaci kaskáda, při vymazání žádná akce, DisciplinaZkratka reference Disciplina (DisciplinaZkratka) při aktualizaci kaskáda, při vymazání žádná akce

VysledekClen	IDVysledek, IDClen, Poradi
Primární klíč	IDVysledek, IDClen
Cizí klíč	IDVysledek reference Vysledek (IDVysledek) při aktualizaci kaskáda, při vymazání kaskáda, IDClen reference Clen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

Disciplina	DisciplinaZkratka, Vzdalenost, Zpusob
Primární klíč	DisciplinaZkratka

3.2.11 Závod

Entita Závod je spojena vztahem N:M s entitou Člen. Znamená to tedy, že člen se účastní více závodů a jednoho závodu se účastní více členů. Vytvoříme tabulku, do níž uložíme kopie primárních klíčů obou entit. Tabulku nazveme UcastikZavodu.

Obě dvě tabulky neporušují ani jednu z prvních tří normálních forem.

Závod	IDZavod, Nazev, Město, Bazen, DatumOd, DatumDo, DelkaBazenu, TypMereni
Primární klíč	IDZavod

UcastnikZavodu	IDZavod, IDClen
Primární klíč	IDZavod, IDClen
Cizí klíč	IDZavod reference Závod (IDZavod) při aktualizaci kaskáda, při vymazání žádná akce, IDClen reference Člen (IDClen) při aktualizaci kaskáda, při vymazání žádná akce

3.2.12 ZaznamOPoskozeni

Vztah s entitou Trener je definován tak, že jeden trenér vytváří více záznamů, ale jeden záznam je vytvořen jedním trenérem. Proto umístíme kopii primárního klíče rodičovské entity Trener do tabulky ZaznamOPoskozeni. Tato entita je ještě ve vztah s entitou Pomucka. Pro spojení tabulek vytvoříme kopii primárního klíče entity Pomucka a umístíme ji do tabulky ZaznamOPoskozeni.

Tabulka je v první, druhé i třetí normální formě, proto nemusíme provádět normalizaci.

ZaznamOPoskozeni	IDZaznam, Nazev, Datum, Popis, IDTrener, IDPomucka
Primární klíč	IDZaznam
Cizí klíč	IDTrener reference Trener (IDTrener) při aktualizaci kaskáda, při vymazání žádná akce, IDPomucka reference Pomucka (IDPomucka) při aktualizaci kaskáda, při vymazání žádná akce

3.2.13 Pomucka

Tato entita je ve vztahu, kromě vztahu popsaného v předchozí části, ještě s entitou Dodavatel. V tomto případě je rodičovskou entitou Dodavatel a proto umístíme kopii primárního klíče této entity do tabulky Pomucka.

Tabulku není nutné normalizovat, neboť je v 1NF, 2NF i 3NF.

Pomucka	IDPomucka, Nazev, Popis, Barva, Počet, DatumNakup, Cena, IDDodavatel
Primární klíč	IDPomucka
Cizí klíč	IDDodavatel reference Dodavatel (IDDodavatel) při aktualizaci kaskáda, při vymazání žádná akce

3.2.14 Dodavatel

Vztah s entitou Pomucka již byl popsán v předchozí části. Primární klíč byl zkopírován do tabulky Pomucka.

Tabulka Dodavatel je v 1NF, 2NF i 3NF. Další normalizaci nemusíme provádět.

Dodavatel	IDDodavatel, Nazev, Telefon, Email, Web
Primární klíč	IDDodavatel

3.2.15 Clen

Všechny vztahy s touto entitou již byly popsány a veškeré cizí klíče potřebné ke spojení tabulek určeny a umístěny do příslušných tabulek.

Tabulka Clen je v první, druhé i třetí normální formě, proto dále nemusíme tabulku normalizovat.

Clen	IDClen, Jmeno, Prijmeni, RodCislo, Ulice, CP, Mesto, PSC, Telefon, Email, DatNar, Pohlavi, SUDaje, SOddil, SInfo, SFoto, DatRegOdd, DatRegSvaz, Plavec
Primární klíč	IDClen
Alternativní klíč	RodCislo; Jmeno+Prijmeni+DatNar+Telefon; Jmeno+Prijmeni+DatNar+Ulice+CP

3.3 Fyzický návrh

Při fyzickém návrhu vycházíme z návrhu logického. Navržené tabulky vytvoříme pomocí příkazů CREATE TABLE. Při vytváření zdrojového kódu určujeme i datové typy podle konkrétního DBMS a vytváříme integritní omezení. Já jsem se rozhodl vytvořit databázi v produktu SQL Server 2008. Zdrojový kód pro vytvoření tabulek a jejich naplnění ukázkovými daty je umístěn v přílohách této práce. Dále jsem vytvořil procedury, které slouží k práci s některými tabulkami.

3.3.1 Procedura VlozClena

Tato jednoduchá procedura slouží k vkládání záznamů o nových členech oddílu. Procedura na základě rodného čísla zadaného uživatelem rozhodne, zdali se vkládaná osoba v databázi nachází či ne. Pokud ano, vypíše chybovou hlášku informaci, pod jakým číslem můžeme záznam v tabulce najít. V opačném případě vytvoří záznam v tabulce a vypíše číslo, které systém přidělil novému členu. Toho je docíleno pomocí příkazu IDENT_CURRENT, který vrátí naposledy generovanou hodnotu identity pro zadanou tabulku.

```
CREATE PROCEDURE VlozClena
@Jmeno varchar(15),
@Prijmeni varchar(20),
@RC char(11),
@Ulice varchar(40),
@CP varchar(6),
@Mesto varchar(40),
@PSC char(5),
@Telefon varchar(14),
@email varchar(60) = NULL,
@DatNar date,
@Pohlavi bit,
@Sudaje bit,
@Soddil bit,
@SInfo bit,
@SFoto bit,
@DatReGOdd date,
@DatRegSvaz date = NULL,
@Plavec bit
AS
BEGIN
    DECLARE @Cislo int
    SELECT @Cislo = IDClen FROM Clen WHERE RodCislo = @RC
    IF (@Cislo is NULL)
    BEGIN
        INSERT INTO Clen (Jmeno,Prijmeni,RodCislo,Ulice,CP,Mesto,PSC,
            Telefon,Email,DatNar,Pohlavi,SUdaje,SOddil,
            SInfo,SFoto,DatReGOdd,DatRegSvaz,Plavec)
```

```

        values
        (@Jmeno,@Prijmeni,@RC,@Ulice,@CP,@Mesto,@PSC,@Telefon,@Email,@DatNar,
        @Pohlavi,@Sudaje,@Soddil,@SInfo,@SFoto,@DatReGOdd,@DatRegSvaz,@Plavec)
        SET @Cislo = IDENT_CURRENT('Clen')
        PRINT 'Osoba byla ulozena. Jeji cislo je: ' + CAST (@Cislo AS
varchar)
    END
    ELSE
    BEGIN
        PRINT 'Osoba se jiz nachazi v databazi! Jeji cislo je: ' + CAST
        (@Cislo AS varchar)
    END
END
GO

```

Procedura se volá pomocí příkazu EXECUTE, za kterým následuje název uložené procedury a její parametry.

```

EXECUTE VlozClena
'Filip','Moudrý','870203/3312','Tichá','78','Znojmo','63100','+4206057
23845','moudry@email.cz','1987-02-03','1','1','1','1','1','1','2013-02-
12',NULL,'1'
GO

```

Při prvním spuštění procedura vypíše hlášení:

```
Osoba byla ulozena. Jeji cislo je: 4
```

Pokud se osoba již v databázi nachází, vypíše procedura:

```
Osoba se jiz nachazi v databazi! Jeji cislo je: 4
```

3.3.2 Procedura CelkemZaObdobi

Tato procedura slouží k zjištění, kolik metrů uplavala zadaná tréninková skupina za určité období. Vstupními parametry procedury jsou identifikátor tréninkové skupiny, počáteční datum a koncové datum. Procedura ověřuje, jestli zadaná tréninková skupina existuje a pokud tomu tak není, vypíše chybovou hlášku. Uživatel je také upozorněn, pokud zadá počáteční datum větší než koncové.

```

CREATE PROCEDURE CelkemZaObdobi
@IDSkupina char(6),
@DatOd date,
@DatDo date
AS
BEGIN
    DECLARE @ID char(6)
    SELECT @ID = IDSkupina FROM TreninkovaSkupina WHERE IDSkupina =
@IDSkupina
    IF (@ID is null)
    BEGIN
        PRINT 'Zadana treninkova skupina neexistuje!'
    END
    ELSE IF (@DatOd > @DatDo)
    BEGIN

```



```

        PRINT 'Zkontrolujte datum!'
    END
    ELSE
        BEGIN
            SELECT IDSkupina, SUM(Delka*Opakovani) AS Celkem FROM
            CastTreninku WHERE Datum >= @DatOd AND Datum <= @DatDo
            AND IDSkupina = @IDSkupina GROUP BY (IDSkupina)
        END
    END
END
GO

```

3.3.3 Procedura CelkemZaObdobiClen

Upravením předchozí procedury získáme proceduru, která vypíše počet uplavaných metrů pro konkrétního člena oddílu. Procedura opět nejprve zkontroluje, zdali člen existuje a pokud tomu tak není, vypíše chybu.

```

CREATE PROCEDURE CelkemZaObdobiClen
@IDClen smallint,
@DatOd date,
@DatDo date
AS
BEGIN
    DECLARE @ID smallint
    SELECT @ID = IDClen FROM Clen WHERE IDClen = @IDClen
    IF (@ID is null)
        BEGIN
            PRINT 'Zadany clen nenalezen v systemu!'
        END
    ELSE IF (@DatOd > @DatDo)
        BEGIN
            PRINT 'Zkontrolujte datum!'
        END
    ELSE
        BEGIN
            DECLARE @Jmeno varchar(15)
            DECLARE @Prijmeni varchar(20)
            DECLARE @Vysledek int
            SELECT @Vysledek = SUM(t.Delka*t.Opakovani) FROM DochazkaTrenink
            c, CastTreninku t WHERE t.Datum >= @DatOd AND t.Datum <= @DatDo AND
            c.IDClen = @IDClen AND c.Ucast = 'U' GROUP BY (c.IDClen)
            SELECT @Jmeno = Clen.Jmeno FROM Clen WHERE Clen.IDClen = @IDClen
            SELECT @Prijmeni = Clen.Prijmeni FROM Clen WHERE Clen.IDClen =
            @IDClen
            PRINT @Jmeno + ' ' + @Prijmeni + ' uplaval/a v zadanem obdobi ' +
            CAST (@Vysledek AS varchar) + ' metru.'
        END
    END
END
GO

EXECUTE CelkemZaObdobiClen '1','2013-05-01','2013-05-31'
GO

```

Pokud zadaný člen existuje a datum je zadáno správně, procedura vypíše hlášení:

Adam Novák uplaval/a v zadanem obdobi 5600 metru.

4 Závěr

Cílem této bakalářské práce bylo vytvoření databáze pro vedení plaveckého oddílu TJ Tesla Brno. Přizpůsobení databáze požadavkům organizace vyžadovalo porozumění procesům probíhajícím uvnitř oddílu a terminologii z oblasti plavání. Hlavními požadavky bylo vytvořit databázovou strukturu tak, aby bylo možné ukládat záznamy o osobách navštěvujících plaveckou výuku, platbách příspěvků, jednotlivých výukových hodinách, materiálu používaném při výuce, závodech a výsledcích ze závodů, záznamy o případných zraněních a záznamy o aktivitách oddílu, které jsou doplňkem hlavní činnosti oddílu.

Oddíl některé informace uchovává v papírové podobě, nebo vůbec ne. Použití databáze přináší výhodu v tom, že data jsou uložena na jednom místě, což do jisté míry zlepšuje jejich dostupnost. Dalším přínosem je, že data uložená v databázi je možné různě analyzovat. Výsledky ze závodů je možné vyhledávat podle zadaných parametrů a snadno je řadit. Náplň tréninku je možné shrnovat dle požadavků uživatele databáze. Spolu se sledováním docházky pak mohou trenéři na základě těchto informací analyzovat výkonnost svých svěřenců.

Pro efektivní práci s databází je ještě nutné navrhnout příslušnou uživatelsky přívětivou aplikaci. Bez jejího použití je práce s databází pro běžného uživatele velmi náročná. Použití online aplikace vytvořené na míru databázi považuji za nejlepší volbu.

Seznam literatury

- 1) CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství – databáze*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- 2) HOTEK, M. *Microsoft SQL Server 2008: Krok za krokem*. Brno: Computer Press, 2009. ISBN 978-80-251-2466-6.
- 3) KOCH, M. a B. NEUWIRTH. *Datové a funkční modelování*. 4. rozšířené vydání. Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, 2010. ISBN 978-80-214-4125-5.
- 4) KŘÍŽ, J. a P. DOSTÁL. *Databázové systémy*. Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, 2005. ISBN 80-214-3064-8.
- 5) MICROSOFT. *Přehled systému Microsoft SQL Server 2008 R2* [online]. Microsoft, ©2012 [cit. 2013-03-10]. Dostupné z: <https://www.microsoft.com/sqlserver/cs/cz/product-info/overview-capabilities.aspx>
- 6) OPPEL, A. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.
- 7) STEPHENS, R., R. PLEW a A. D. JONES. *Naučte se SQL za 28 dní*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.
- 8) TJ TESLA BRNO. *Stanovy TJ TESLA BRNO*. Brno: TJ TESLA BRNO, 2013.
- 9) TJ TESLA BRNO. *TJ Tesla Brno* [online]. TJ TESLA BRNO, ©2004-2012 [cit. 2013-03-06]. Dostupné z: <http://www.tjtesla.cz/>

Seznam obrázků

Organizační struktura vedení plaveckého oddílu (Zdroj: Vlastní zpracování).....	24
ER diagram - první verze (Zdroj: vlastní zpracování).....	33
Entita Clen (Zdroj: vlastní zpracování)	36
Entita Trener (Zdroj: vlastní zpracování)	36
Entita Kvalifikace (Zdroj: vlastní zpracování)	37
Entita Platba (Zdroj: vlastní zpracování)	37
Entita ClenskaKarta (Zdroj: vlastní zpracování)	38
Entita TreninkovaSkupina (Zdroj: vlastní zpracování)	39
Entita Trenink (Zdroj: vlastní zpracování)	39
Entita CastTreninku (Zdroj: vlastní zpracování)	40
Entita Aktivita (Zdroj: vlastní zpracování).....	40
Entita ZaznamOZraneni (Zdroj: vlastní zpracování).....	41
Entita Zavod (Zdroj: vlastní zpracování).....	42
Entita Vysledek (Zdroj: vlastní zpracování).....	42
Entita Pomucka (Zdroj: vlastní zpracování)	43
Entita Dodavatel (Zdroj: vlastní zpracování).....	43
Entita ZaznamOPoskozeni (Zdroj: vlastní zpracování).....	44
ER diagram - konceptuální návrh (Zdroj: vlastní zpracování)	45

Seznam tabulek

Výuka plavání v TJ Tesla Brno k 1.1.2013 (Zdroj: vlastní zpracování)	26
Seznam entit (Zdroj: Vlastní zpracování).....	32
Vztahy mezi entitami (Zdroj: vlastní zpracování)	34

Seznam příloh

Příloha č. 1 – Datový slovník

Příloha č. 2 – Schéma databáze

Příloha č. 4 – Zdrojový kód v SQL

Příloha č. 3 – CD se zdrojovým kódem v SQL

Datový slovník

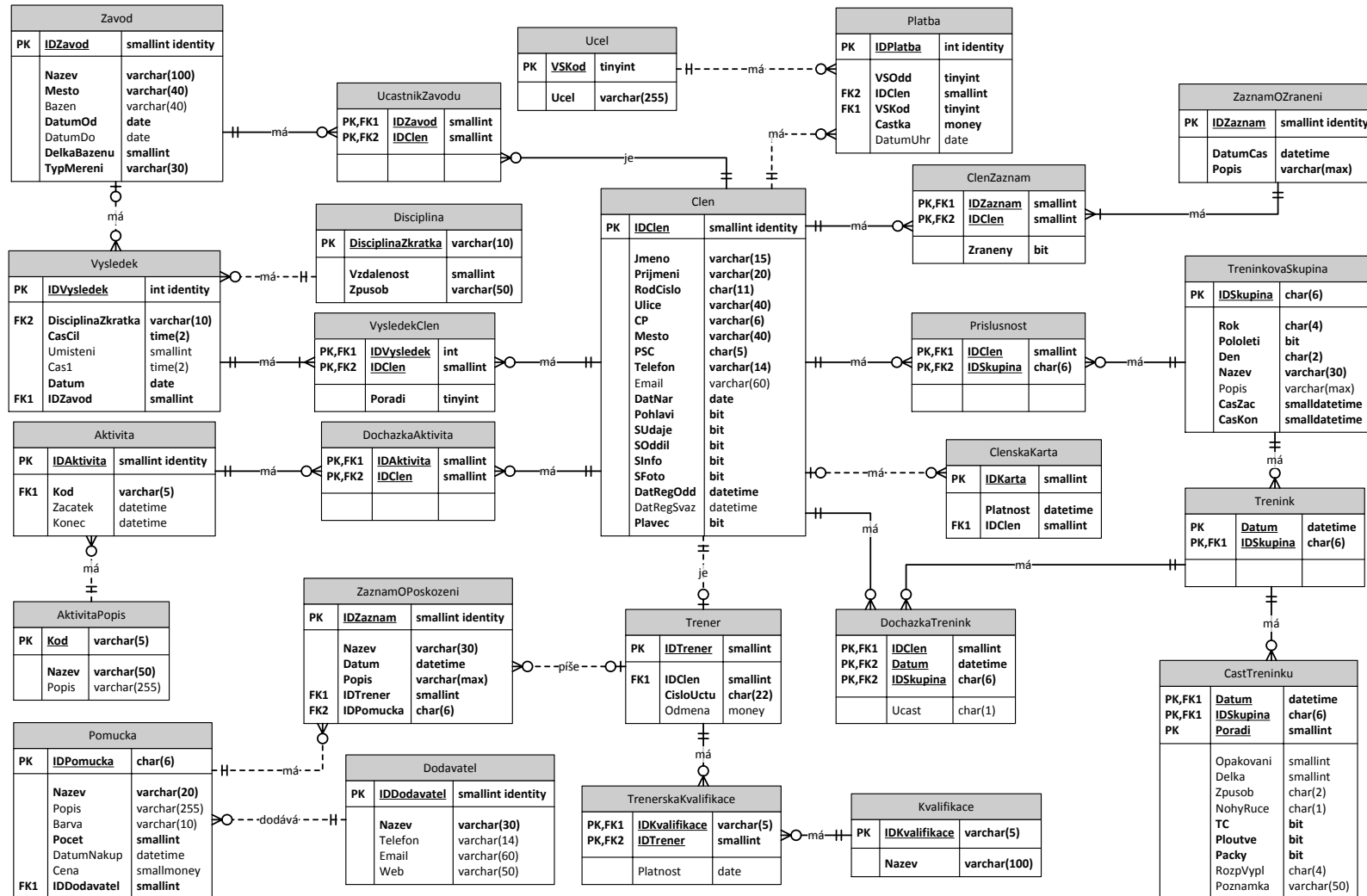
Entita	Jméno atributu	Popis	Typ a délka	NULL	PK
Trener	IDTrener	Jednoznačná identifikace trenéra	číslo	Ne	PK
	Jmeno	Křestní jméno zaměstnance	15 znaků, proměnlivá délka	Ne	
	Prijmeni	Příjmení zaměstnance	20 znaků, proměnlivá délka	Ne	
	CisloUctu	Číslo bankovního účtu	22 znaků, pevná délka	Ne	
	Odmena	Výše hodinové mzdy v Kč	měna v Kč	Ano	
Clen	IDClen	Jednoznačná identifikace člena	číslo	Ne	PK
	Jmeno	Křestní jméno člena	15 znaků, proměnlivá délka	Ne	
	Prijmeni	Příjmení člena	20 znaků, proměnlivá délka	Ne	
	RodCislo	Rodné číslo člena	11 znaků, pevná délka	Ne	
	Ulice	Název ulice v bydlišti člena	40 znaků, proměnlivá délka	Ne	
	CP	Popisné číslo budovy	6 znaků, proměnlivá délka	Ne	
	Mesto	Název města	40 znaků, proměnlivá délka	Ne	
	PSC	Poštovní směrovací číslo	5 znaků, pevná délka	Ne	
	Telefon	Telefonní číslo na člena	14 znaků, proměnlivá délka	Ne	
	Email	E-mailová adresa na člena	60 znaků, proměnlivá délka	Ano	
	DatNar	Datum narození	datum, lze získat z RodCislo	Ne	
	Pohlavi	Pohlaví člena	logická hodnota	Ne	
	SUdaje	Podepsané prohlášení o souhlasu se zpracováním a evidencí osobních údajů	logická hodnota	Ne	
	SOddil	Souhlas s podmínkami plaveckého oddílu	logická hodnota	Ne	
	SInfo	Souhlas se zasíláním informací o činnosti oddílu	logická hodnota	Ne	
	SFoto	Souhlas s uveřejněním fotografií na webových stránkách	logická hodnota	Ne	
	DatRegOdd	Datum registrace do oddílu	datum	Ne	
	DatRegSvaz	Datum registrace na svazu plavání	datum	Ano	
	Plavec	Člen plaveckého oddílu	logická hodnota	Ne	

Entita	Jméno atributu	Popis	Typ a délka	NULL	PK
ClenskaKarta	IDKarty	Identifikační číslo karty	číslo	Ne	PK
	Jmeno	Křestní jméno vlastníka karty	15 znaků, proměnlivá délka	Ne	
	Prijmeni	Příjmení vlastníka karty	20 znaků, proměnlivá délka	Ne	
	Platnost	Platnost členské karty	datum	Ne	
Trenink	Datum	Datum konání tréninku	datum	Ne	PK
	IDSkupina	Identifikace tréninkové skupiny	6 znaků, pevná délka	Ne	PK
Zavod	IDZavod	Jednoznačná identifikace závodu	číslo	Ne	PK
	Nazev	Název závodu	100 znaků, proměnlivá délka	Ne	
	Mesto	Název města, kde se koná závod	40 znaků, proměnlivá délka	Ne	
	Bazen	Název bazénu či jeho adresa	40 znaků, proměnlivá délka	Ano	
	DatumOd	Datum začátku závodů	datum	Ne	
	DatumDo	Datum konce závodů	datum	Ano	
	DelkaBazenu	Délka bazénu v metrech	číslo	Ne	
	TypMereni	Typ měření výsledného času	30 znaků, proměnlivá délka	Ne	
Pomucka	IDPomucka	Jednoznačné určení pomůcky	6 znaků, pevná délka	Ne	PK
	Nazev	Název pomůcky	20 znaků, proměnlivá délka	Ne	
	Popis	Popis upřesňující o jakou pomůcku jde	255 znaků, proměnlivá délka	Ano	
	Barva	Barva pomůcky	10 znaků, proměnlivá délka	Ano	
	Pocet	Počet pomůcek	číslo	Ne	
	DatNakup	Datum pořízení pomůcky	datum	Ano	
	Cena	Cena za jednotku	měna v Kč	Ano	
Aktivita	IDAktivita	Jednoznačně identifikuje aktivitu	číslo	Ne	PK
	Kod	Kódové označení aktivity	5 znaků, proměnlivá délka	Ne	
	Nazev	Název aktivity	50 znaků, proměnlivá délka	Ne	
	Popis	Detailnější popis aktivity	255 znaků, proměnlivá délka	Ano	
	DatCasOd	Datum a čas začátku	datum a čas	Ne	
	DatCasKon	Datum a čas konce	datum a čas	Ano	

Entita	Jméno atributu	Popis	Typ a délka	NULL	PK
Platba	IDPlatba	Jednoznačná identifikace platby	číslo	Ne	PK
	VarSymbol	Variabilní symbol platby	8 znaků, pevná délka	Ne	
	Ucel	Popisuje účel platby	255 znaků, proměnlivá délka	Ne	
	Castka	Výše částky k úhradě	měna v Kč	Ne	
	DatumUhr	Datum splacení částky	datum	Ano	
Kvalifikace	IDKvalifikace	Zkratka jednoznačně identifikující danou kvalifikaci	5 znaků, proměnlivá délka	Ne	PK
	Název	Název kvalifikace	100 znaků, proměnlivá délka	Ne	
TreninkovaSkupina	IDSkupina	Jednoznačná identifikace skupiny	6 znaků, pevná délka	Ne	PK
	Rok	Rok, ve kterém se skupina zúčastňuje výuky	číslo	Ne	
	Pololeti	Pololetí roku	logická hodnota (0 – první, 1 – druhé pololetí)	Ne	
	Den	Den v týdnu, kdy má skupina výuku	2 znaky, pevná délka	Ne	
	Nazev	Název skupiny	30 znaků, proměnlivá hodnota	Ne	
	Popis	Popisuje podrobněji skupinu	text	Ano	
	CasZac	Čas začátku hodiny	čas	Ne	
	CasKon	Čas konce hodiny	čas	Ne	
CastTreninku	Datum	Datum konání tréninku	datum	Ne	PK
	IDSkupina	Označení tréninkové skupiny	6 znaků, pevná délka	Ne	PK
	Poradi	Pořadí části tréninku	číslo	Ne	PK
	Opakovani	Zadáva se při opakování stejné části	číslo	Ano	
	Delka	Délka části v metrech	číslo	Ano	
	Zpusob	Zkratka plaveckého způsobu	2 znaky, proměnlivá délka	Ano	
	NohyRuce	Plavání s deskou, používají se jen nohy nebo ruce	1 znak, pevná délka	Ano	
	TC	Plavání technickým cvičením	logická hodnota	Ne	
	Ploutve	Plavání s ploutvemi	logická hodnota	Ne	
	Packy	Plavání s packami	logická hodnota	Ne	
	RozplVypl	Označení zda jde o rozplavání či vyplavání	4 znaky, pevná délka	Ano	
	Poznamka	Poznámka slouží například pro zadání o tempu	50 znaků, proměnlivá délka	Ano	

Entita	Jméno atributu	Popis	Typ a délka	NULL	PK
ZaznamOZraneni	IDZaznam	Jednoznačná identifikace každého incidentu	číslo	Ne	PK
	DatumCas	Datum a čas, kdy k incidentu došlo	datum a čas	Ne	
	Popis	Popis incidentu	text	Ne	
	JmenoZraneny	Jméno zraněného člena	složený atribut (jméno, příjmení)	Ne	
	JmenoTrener	Jméno trenéra, který incident řešil	složený atribut (jméno, příjmení)	Ne	
Vysledek	IDVysledek	Jednoznačné označení výsledku	číslo	Ne	PK
	Disciplina	Název disciplíny	složený atribut (vzdálenost, způsob)	Ne	
	Jmeno1	Jméno závodníka / prvního člena štafety	15 znaků, proměnlivá hodnota	Ne	
	Prijmeni1	Příjmení závodníka / prvního člena štafety	20 znaků, proměnlivá délka	Ne	
	CasCil	Výsledný čas plavce / štafety	čas	Ne	
	Umisteni	Dosažené umístění plavce / štafety	číslo	Ano	
	Jmeno2	Jméno druhého závodníka při štafetě	složený atribut (jméno, příjmení)	Ano	
	Jmeno3	Jméno třetího závodníka při štafetě	složený atribut (jméno, příjmení)	Ano	
	Jmeno4	Jméno čtvrtého závodníka při štafetě	složený atribut (jméno, příjmení)	Ano	
	Cas1	Čas prvního člena štafety	čas	Ano	
	Datum	Datum dosažení výkonu	datum	Ne	
ZaznamOPoskozeni	IDZaznam	Jednoznačná identifikace záznamu	číslo	Ne	PK
	Nazev	Název pomůcky	30 znaků, proměnlivá délka	Ne	
	Datum	Datum zápisu poškození	datum	Ne	
	Popis	Popis rozsahu a typu poškození	text	Ne	
Dodavatel	IDDodavatel	Jednoznačná identifikace dodavatele	číslo	Ne	PK
	Nazev	Název dodavatele	30 znaků, proměnlivá délka	Ne	
	Telefon	Telefonní číslo na dodavatele	14 znaků, proměnlivá délka	Ano	
	Email	E-mailová adresa dodavatele	60 znaků, proměnlivá délka	Ano	
	Web	Webové stránky dodavatele	50 znaků, proměnlivá délka	Ano	

Schéma databáze



Zdrojový kód

```
-- vytvoreni tabulek
CREATE TABLE Clen (
    IDClen smallint identity (1,1),
    Jmeno varchar(15) NOT NULL,
    Prijmeni varchar(20) NOT NULL,
    RodCislo char(11) NOT NULL,
    Ulice varchar(40) NOT NULL,
    CP varchar(6) NOT NULL,
    Mesto varchar(40) NOT NULL,
    PSC char(5) NOT NULL,
    Telefon varchar(14) NOT NULL,
    Email varchar(60),
    DatNar date NOT NULL,
    Pohlavi bit NOT NULL,
    SUDaje bit NOT NULL DEFAULT '0',
    SOddil bit NOT NULL DEFAULT '0',
    SInfo bit NOT NULL DEFAULT '0',
    SFoto bit NOT NULL DEFAULT '0',
    DatReGOdd date NOT NULL,
    DatRegSvaz date,
    Plavec bit NOT NULL DEFAULT '0',
    PRIMARY KEY (IDClen))
GO

CREATE TABLE Trener (
    IDTrener smallint CHECK (IDTrener >= 0),
    CisloUctu char(22) NOT NULL,
    Odmena money DEFAULT 0 CHECK (Odmena >= 0),
    IDClen smallint CHECK (IDClen > 0),
    PRIMARY KEY (IDTrener),
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

CREATE TABLE Ucel (
    VSKod tinyint CHECK (VSKod > 0 AND VSKod <= 99),
    Ucel varchar(255) NOT NULL,
    PRIMARY KEY (VSKod))
GO

CREATE TABLE Platba (
    IDPlatba int identity(1,1),
    VSOdd tinyint DEFAULT '16' NOT NULL CHECK (VSOdd >= 0 AND VSOdd <=
99),
    Castka money NOT NULL CHECK (Castka >= 0),
    DatumUhr date,
    IDClen smallint CHECK (IDClen > 0),
    VSKod tinyint CHECK (VSKod > 0 AND VSKod <= 99),
    PRIMARY KEY (IDPlatba),
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE,
    FOREIGN KEY (VSKod) REFERENCES Ucel (VSKod) ON UPDATE CASCADE)
GO

CREATE TABLE Kvalifikace (
    IDKvalifikace varchar(5),
    Nazev varchar(100) NOT NULL,
    PRIMARY KEY (IDKvalifikace))
```

GO

```
CREATE TABLE TrenerskaKvalifikace (  
    Platnost date,  
    IDKvalifikace varchar(5),  
    IDTrener smallint CHECK (IDTrener > 0),  
    PRIMARY KEY (IDKvalifikace, IDTrener),  
    FOREIGN KEY (IDKvalifikace) REFERENCES Kvalifikace (IDKvalifikace)  
ON UPDATE CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (IDTrener) REFERENCES Trener (IDTrener) ON UPDATE  
CASCADE ON DELETE CASCADE)  
GO
```

```
CREATE TABLE Dodavatel (  
    IDDodavatel smallint identity(1,1),  
    Nazev varchar(30) NOT NULL,  
    Telefon varchar(14),  
    Email varchar(60),  
    Web varchar(50),  
    PRIMARY KEY (IDDodavatel))  
GO
```

```
CREATE TABLE Pomucka (  
    IDPomucka char(6),  
    Nazev varchar(20) NOT NULL,  
    Popis varchar(255),  
    Barva varchar(10),  
    Pocet smallint NOT NULL DEFAULT '1' CHECK (Pocet >= 0),  
    DatumNakupu date,  
    Cena money CHECK (Cena >= 0),  
    IDDodavatel smallint CHECK (IDDodavatel > 0),  
    PRIMARY KEY (IDPomucka),  
    FOREIGN KEY (IDDodavatel) REFERENCES Dodavatel (IDDodavatel) ON  
UPDATE CASCADE)  
GO
```

```
CREATE TABLE ZaznamOPoskozeni (  
    IDZaznam smallint identity(1,1),  
    Nazev varchar(30) NOT NULL,  
    Datum datetime NOT NULL,  
    Popis text,  
    IDTrener smallint CHECK (IDTrener > 0),  
    IDPomucka char(6),  
    PRIMARY KEY (IDZaznam),  
    FOREIGN KEY (IDTrener) REFERENCES Trener (IDTrener) ON UPDATE  
CASCADE,  
    FOREIGN KEY (IDPomucka) REFERENCES Pomucka (IDPomucka) ON UPDATE  
CASCADE)  
GO
```

```
CREATE TABLE AktivitaPopis (  
    Kod varchar(5),  
    Nazev varchar(50) NOT NULL,  
    Popis varchar(255),  
    PRIMARY KEY (Kod))  
GO
```

```
CREATE TABLE Aktivita (  
    IDAktivita smallint identity(1,1),
```

```

Kod varchar(5),
Zacatek datetime NOT NULL,
Konec datetime,
PRIMARY KEY (IDAktivita),
FOREIGN KEY (Kod) REFERENCES AktivitaPopis (Kod) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE DochazkaAktivita (
    IDAktivita smallint CHECK (IDAktivita > 0),
    IDClen smallint CHECK (IDClen > 0),
    PRIMARY KEY (IDAktivita, IDClen),
    FOREIGN KEY (IDAktivita) REFERENCES Aktivita (IDAktivita) ON UPDATE CASCADE,
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE Disciplina (
    DisciplinaZkratka varchar(10),
    Vzdalenost smallint NOT NULL CHECK (Vzdalenost > 0),
    Zpusob varchar(50) NOT NULL,
    PRIMARY KEY (DisciplinaZkratka))
GO

```

```

CREATE TABLE Zavod (
    IDZavod smallint identity(1,1),
    Nazev varchar(100) NOT NULL,
    Mesto varchar(40) NOT NULL,
    Bazenu varchar(40),
    DatumOd date NOT NULL,
    DatumDo date,
    DelkaBazenu smallint NOT NULL CHECK (DelkaBazenu > 0),
    TypMereni varchar(30) NOT NULL,
    PRIMARY KEY (IDZavod))
GO

```

```

CREATE TABLE UcastnikZavodu (
    IDZavod smallint CHECK (IDZavod > 0),
    IDClen smallint CHECK (IDClen > 0),
    PRIMARY KEY (IDZavod, IDClen),
    FOREIGN KEY (IDZavod) REFERENCES Zavod (IDZavod) ON UPDATE CASCADE,
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE Vysledek (
    IDVysledek int identity(1,1),
    DisciplinaZkratka varchar(10) NOT NULL,
    CasCil time(2) NOT NULL,
    Umisteni smallint CHECK (Umisteni > 0),
    Cas1 time(2),
    Datum date,
    IDZavod smallint CHECK (IDZavod > 0),
    PRIMARY KEY (IDVysledek),
    FOREIGN KEY (DisciplinaZkratka) REFERENCES Disciplina
(DisciplinaZkratka) ON UPDATE CASCADE,
    FOREIGN KEY (IDZavod) REFERENCES Zavod (IDZavod) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE VysledekClen (
    IDVysledek int CHECK (IDVysledek > 0),

```

```

    IDClen smallint CHECK (IDClen > 0),
    Poradi tinyint NOT NULL CHECK (Poradi > 0 AND Poradi <= 4),
    PRIMARY KEY (IDVysledek, IDClen),
    FOREIGN KEY (IDVysledek) REFERENCES Vysledek (IDVysledek) ON UPDATE
CASCADE ON DELETE CASCADE,
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE ZaznamOZraneni (
    IDZaznam smallint identity(1,1),
    DatumCas datetime NOT NULL,
    Popis text NOT NULL,
    PRIMARY KEY (IDZaznam))
GO

```

```

CREATE TABLE ClenZaznam (
    IDZaznam smallint CHECK (IDZaznam > 0),
    IDClen smallint CHECK (IDClen > 0),
    Zraneny bit NOT NULL,
    PRIMARY KEY (IDZaznam, IDClen),
    FOREIGN KEY (IDZaznam) REFERENCES ZaznamOZraneni (IDZaznam) ON
UPDATE CASCADE,
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

```

```

CREATE TABLE ClenskaKarta (
    IDKarta smallint CHECK (IDKarta > 0),
    Platnost date NOT NULL,
    IDClen smallint CHECK (IDClen > 0),
    PRIMARY KEY (IDKarta),
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE ON
DELETE CASCADE)
GO

```

```

CREATE TABLE TreninkovaSkupina (
    IDSkupina char(6),
    Rok smallint NOT NULL CHECK (Rok > 0),
    Pololeti bit NOT NULL,
    Den char(2) NOT NULL,
    Nazev varchar(30) NOT NULL,
    Popis text,
    CasZac smalldatetime NOT NULL,
    CasKon smalldatetime NOT NULL,
    PRIMARY KEY (IDSkupina))
GO

```

```

CREATE TABLE Prislusnost (
    IDClen smallint CHECK (IDClen > 0),
    IDSkupina char(6),
    PRIMARY KEY (IDClen, IDSkupina),
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY (IDSkupina) REFERENCES TreninkovaSkupina (IDSkupina) ON
UPDATE CASCADE)
GO

```

```

CREATE TABLE Trenink (
    Datum date,
    IDSkupina char(6),

```

```

    PRIMARY KEY (Datum, IDSkupina),
    FOREIGN KEY (IDSkupina) REFERENCES TreninkovaSkupina (IDSkupina) ON
UPDATE CASCADE)
GO

CREATE TABLE DochazkaTrenink (
    IDClen smallint CHECK (IDClen > 0),
    IDSkupina char(6),
    Datum date,
    Ucast char(1) NOT NULL,
    PRIMARY KEY (IDClen, IDSkupina, Datum),
    FOREIGN KEY (Datum, IDSkupina) REFERENCES Trenink (Datum, IDSkupina)
ON UPDATE CASCADE,
    FOREIGN KEY (IDClen) REFERENCES Clen (IDClen) ON UPDATE CASCADE)
GO

CREATE TABLE CastTreninku (
    IDSkupina char(6),
    Datum date,
    Poradi smallint CHECK (Poradi > 0),
    Opakovani smallint NOT NULL DEFAULT '1' CHECK (Opakovani > 0),
    Delka smallint CHECK (Delka > 0),
    Zpusob char(2),
    NohyRuce char(1),
    TC bit NOT NULL DEFAULT '0',
    Ploutve bit NOT NULL DEFAULT '0',
    Packy bit NOT NULL DEFAULT '0',
    RozpVyp1 char(4),
    Poznamka varchar(50),
    PRIMARY KEY (IDSkupina, Datum, Poradi),
    FOREIGN KEY (Datum, IDSkupina) REFERENCES Trenink (Datum, IDSkupina)
ON UPDATE CASCADE ON DELETE CASCADE)
GO

-- naplneni tabulek vzorovymi daty
INSERT INTO Clen
(Jmeno, Prijmeni, RodCislo, Ulice, CP, Mesto, PSC, Telefon, Email, DatNar, Pohla
vi, SUdaje, SOddil, SInfo, SFoto, DatReGOdd, DatRegSvaz, Plavec)
VALUES
('Adam', 'Novák', '890606/9878', 'Dlouhá', '123', 'Brno', '61200', '+42077712
3456', 'novak@email.cz', '1989-06-06', '1', '1', '1', '1', '1', '2012-01-
23', NULL, '1')
INSERT INTO Clen
(Jmeno, Prijmeni, RodCislo, Ulice, CP, Mesto, PSC, Telefon, Email, DatNar, Pohla
vi, SUdaje, SOddil, SInfo, SFoto, DatReGOdd, DatRegSvaz, Plavec)
VALUES
('Adéla', 'Nováková', '895606/9878', 'Dlouhá', '123', 'Brno', '61200', '+4207
77123456', 'novakova@email.cz', '1989-06-06', '0', '1', '1', '1', '1', '2012-
01-23', NULL, '1')
INSERT INTO Clen
(Jmeno, Prijmeni, RodCislo, Ulice, CP, Mesto, PSC, Telefon, Email, DatNar, Pohla
vi, SUdaje, SOddil, SInfo, SFoto, DatReGOdd, DatRegSvaz, Plavec)
VALUES ('Karel', 'Malý', '820306/1878', 'Milady
Horákové', '13', 'Krnov', '65202', '+420603123456', 'maly@email.cz', '1982-
03-06', '1', '1', '1', '1', '1', '2010-01-23', NULL, '1')
GO

SELECT * FROM Clen
GO

```

```
INSERT INTO Trener VALUES ('0001','000000-1347894567/0300','120','3')
GO
```

```
SELECT * FROM Trener
GO
```

```
INSERT INTO ClenskaKarta VALUES ('1','2013-12-31','3')
INSERT INTO ClenskaKarta VALUES ('2','2013-12-31','2')
INSERT INTO ClenskaKarta VALUES ('3','2013-12-31','1')
GO
```

```
SELECT * FROM ClenskaKarta
GO
```

```
INSERT INTO TreninkovaSkupina VALUES
('1312N0','2013','0','Út','Neplavci','Skupina urcena
neplavcum','17:00','17:30')
GO
```

```
SELECT * FROM TreninkovaSkupina
GO
```

```
INSERT INTO Prislusnost VALUES ('1','1312N0')
INSERT INTO Prislusnost VALUES ('2','1312N0')
INSERT INTO Prislusnost VALUES ('3','1312N0')
GO
```

```
SELECT * FROM Prislusnost
GO
```

```
INSERT INTO Trenink VALUES ('2013-05-14','1312N0')
INSERT INTO Trenink VALUES ('2013-05-07','1312N0')
INSERT INTO Trenink VALUES ('2013-04-07','1312N0')
GO
```

```
SELECT * FROM Trenink
GO
```

```
INSERT INTO DochazkaTrenink (IDClen,IDSkupina,Datum,Ucast) VALUES
('1','1312N0','2013-05-07','U')
INSERT INTO DochazkaTrenink (IDClen,IDSkupina,Datum,Ucast) VALUES
('2','1312N0','2013-05-07','U')
INSERT INTO DochazkaTrenink (IDClen,IDSkupina,Datum,Ucast) VALUES
('3','1312N0','2013-05-07','U')
INSERT INTO DochazkaTrenink (IDClen,IDSkupina,Datum,Ucast) VALUES
('1','1312N0','2013-05-14','U')
INSERT INTO DochazkaTrenink (IDClen,IDSkupina,Datum,Ucast) VALUES
('3','1312N0','2013-05-14','U')
GO
```

```
SELECT * FROM DochazkaTrenink
GO
```

```
INSERT INTO CastTreninku (IDSkupina,Datum,Poradi,Delka,RozpVyp1)
VALUES ('1312N0','2013-05-07','1','100','rozp')
INSERT INTO CastTreninku
(IDSkupina,Datum,Poradi,Opakovani,Delka,Zpusob,NohyRuce,Ploutve)
VALUES ('1312N0','2013-05-07','2','16','50','VZ','N','1')
```



```

INSERT INTO CastTreninku (IDSkupina,Datum,Poradi,Delka,RozpVypl)
VALUES ('1312N0','2013-05-07','3','100','vyp1')
INSERT INTO CastTreninku (IDSkupina,Datum,Poradi,Delka,RozpVypl)
VALUES ('1312N0','2013-05-14','1','100','rozp')
INSERT INTO CastTreninku
(IDSkupina,Datum,Poradi,Opakovani,Delka,Zpusob) VALUES
('1312N0','2013-05-14','2','16','100','P')
INSERT INTO CastTreninku (IDSkupina,Datum,Poradi,Delka,RozpVypl)
VALUES ('1312N0','2013-05-14','3','100','vyp1')
INSERT INTO CastTreninku (IDSkupina,Datum,Poradi,Delka,RozpVypl)
VALUES ('1312N0','2013-04-07','1','100','rozp')
GO

SELECT * FROM CastTreninku
GO

-- vytvoreni procedur
CREATE PROCEDURE VlozClena
@Jmeno varchar(15),
@Prijmeni varchar(20),
@RC char(11),
@Ulice varchar(40),
@CP varchar(6),
@Mesto varchar(40),
@PSC char(5),
@Telefon varchar(14),
@email varchar(60) = NULL,
@DatNar date,
@Pohlavi bit,
@Sudaje bit,
@Soddil bit,
@SInfo bit,
@SFoto bit,
@DatReGOdd date,
@DatRegSvaz date = NULL,
@Plavec bit
AS
BEGIN
    DECLARE @Cislo int
    SELECT @Cislo = IDClen FROM Clen WHERE RodCislo = @RC
    IF (@Cislo is NULL)
        BEGIN
            INSERT INTO Clen (Jmeno,Prijmeni,RodCislo,Ulice,CP,Mesto,PSC,
                             Telefon,Email,DatNar,Pohlavi,SUdaje,SODdil,
                             SInfo,SFoto,DatReGOdd,DatRegSvaz,Plavec)
                values
                (@Jmeno,@Prijmeni,@RC,@Ulice,@CP,@Mesto,@PSC,@Telefon,@Email,@DatNar,
                @Pohlavi,@Sudaje,@Soddil,@SInfo,@SFoto,@DatReGOdd,@DatRegSvaz,@Plavec)
            SET @Cislo = IDENT_CURRENT('Clen')
            PRINT 'Osoba byla ulozena. Jeji cislo je: ' + CAST (@Cislo AS
varchar)
        END
    ELSE
        BEGIN
            PRINT 'Osoba se jiz nachazi v databazi! Jeji cislo je: ' + CAST
(@Cislo AS varchar)
        END
END
GO

```

```

CREATE PROCEDURE CelkemZaObdobi
@IDSkupina char(6),
@DatOd date,
@DatDo date
AS
BEGIN
    DECLARE @ID char(6)
    SELECT @ID = IDSkupina FROM TreninkovaSkupina WHERE IDSkupina =
@IDSkupina
    IF (@ID is NULL)
    BEGIN
        PRINT 'Zadana treninkova skupina neexistuje!'
    END
    ELSE IF (@DatOd > @DatDo)
    BEGIN
        PRINT 'Zkontrolujte datum!'
    END
    ELSE
    BEGIN
        SELECT IDSkupina, SUM(Delka*Opakovani) AS Celkem FROM
CastTreninku WHERE Datum >= @DatOd AND Datum <= @DatDo
AND IDSkupina = @IDSkupina GROUP BY (IDSkupina)
    END
END
GO

CREATE PROCEDURE CelkemZaObdobiClen
@IDClen smallint,
@DatOd date,
@DatDo date
AS
BEGIN
    DECLARE @ID smallint
    SELECT @ID = IDClen FROM Clen WHERE IDClen = @IDClen
    IF (@ID is NULL)
    BEGIN
        PRINT 'Zadany clen nenalezen v systemu!'
    END
    ELSE IF (@DatOd > @DatDo)
    BEGIN
        PRINT 'Zkontrolujte datum!'
    END
    ELSE
    BEGIN
        DECLARE @Jmeno varchar(15)
        DECLARE @Prijmeni varchar(20)
        DECLARE @Vysledek int
        SELECT @Vysledek = SUM(t.Delka*t.Opakovani) FROM DochazkaTrenink
c, CastTreninku t WHERE t.Datum >= @DatOd AND t.Datum <= @DatDo AND
c.IDClen = @IDClen AND c.Ucast = 'U' GROUP BY (c.IDClen)
        SELECT @Jmeno = Clen.Jmeno FROM Clen WHERE Clen.IDClen = @IDClen
        SELECT @Prijmeni = Clen.Prijmeni FROM Clen WHERE Clen.IDClen =
@IDClen
        PRINT @Jmeno + ' ' + @Prijmeni + ' uplaval/a v zadanem obdobi ' +
CAST (@Vysledek AS varchar) + ' metru.'
    END
END
GO

```

```
-- volani procedur
EXECUTE VlozClena
'Filip','Moudrý','870203/3312','Tichá','78','Znojmo','63100','+4206057
23845','moudry@email.cz','1987-02-03','1','1','1','1','1','2013-02-
12',NULL,'1'
GO

EXECUTE CelkemZaObdobi '1312N0','2013-05-01','2013-05-31'
GO

EXECUTE CelkemZaObdobiClen '1','2013-05-01','2013-05-31'
GO
```