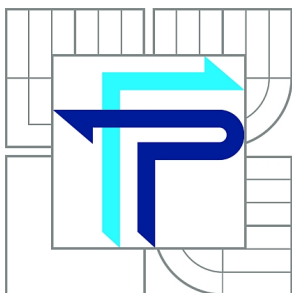




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE SQL PRO MORAVSKÉ GYMNÁZIUM BRNO S.R.O.

PROPOSAL OF SQL DATABASE FOR MORAVSKÉ GYMNAZIUM BRNO S.R.O.

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ KIRSCH

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ KŘÍŽ, Ph.D.

BRNO 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Kirsch Jiří

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:

Návrh databáze SQL pro Moravské gymnázium Brno s.r.o.

v anglickém jazyce:

Proposal of SQL Database for Moravské gymnázium Brno s.r.o.

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., BEGG, C., HOLOWCZAK, R. Mistrovství – databáze: Profesionální průvodce tvorbou efektivních databází. 1. vydání. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
- GROFF, J., WEINBERG, P. SQL Kompletní průvodce. 1. vydání. Brno: Computer Press, 2005. 936 s. ISBN 80-251-0369-2.
- HOTEK, M. Microsoft SQL Server 2008: krok za krokem. 1.vydání. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6.
- HOULETTE, F. SQL: Příručka programátora. 1. vydání. Praha: SoftPress, 2001. 382 s. ISBN 80-86497-14-3.
- PUŽMANOVÁ, R. Moderní komunikační sítě od A do Z. 2. aktualizované vydání. Brno: Computer Press, 2006. 430 s. ISBN 80-251-1278-0.

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

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Děkan fakulty

V Brně, dne 19.05.2012

Abstrakt

Obsahem této bakalářské práce je návrh SQL databáze pro společnost Moravské gymnázium Brno s.r.o. Práce je rozdělena na teoretickou část a praktickou část. V části první si uvedeme potřebnou teorii pro úspěšné dosažení cílů práce. Praktickou část tvoří analýza současné situace ve firmě pro zmíněnou problematiku a návrh na její řešení.

Abstract

The content of this bachelor's thesis is Proposal of SQL Database for Moravské gymnázium Brno s.r.o. Work is divided into theoretical and practical part. In first part I will describe the theoretical basics for successful achievement of goals. Practical part is created by analysis of present situation and own proposal of database.

Klíčová slov

databáze, SQL, škola, normalizace, dotaz, procedura

Key words

database, SQL, school, normalization, query, procedure

Bibliografická citace práce

KIRSCH, J. *Návrh databáze SQL pro Moravské gymnázium Brno*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2012. 68 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 v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb. o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 22. 5. 2012

Jiří Kirsch

Poděkování

Rád bych tímto poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Křížovi, Ph.D. za vstřícný přístup, cenné rady a připomínky, které mi pomohly při řešení této práce.

Obsah

Úvod.....	10
1 Vymezení problému a cíl práce	11
2 Teoretická východiska	12
2.1 Data, informace a znalosti.....	12
2.2 Databáze.....	13
2.2.1 Lineární datový model	13
2.2.2 Hierarchický datový model.....	13
2.2.3 Síťový datový model.....	14
2.2.4 Objektový datový model.....	14
2.2.5 Relační datový model	14
2.3 Terminologie	15
2.3.1 Entita.....	15
2.3.2 Vazby mezi entitami	15
2.3.3 Klíče tabulek	16
2.3.4 Relační integrita.....	16
2.3.5 Normalizace	16
2.4 Tvorba návrhu databáze	17
2.4.1 Životní cyklus databáze	18
2.4.2 Konceptuální návrh.....	18
2.4.3 Logický návrh	20
2.4.4 Fyzický návrh	20
2.5 SQL (Structured Query Language)	21
2.5.1 Základní dělení SQL.....	21
2.5.2 Datové typy v SQL	22

3	Analýza problému a současného stavu	24
3.1	Základní údaje společnosti	24
3.2	Analýza současné situace	26
3.3	Informační technologie gymnázia.....	26
3.3.1	Hardwarové vybavení	27
3.3.2	Softwarové vybavení	28
3.4	Školní informační systémy.....	28
3.4.1	SAS	29
3.4.2	Škola OnLine	29
3.4.3	Bakaláři.....	30
3.4.4	Souhrnný přehled.....	31
4	Návrh vlastního řešení, přínos návrhu	32
4.1	Požadavky na databázi	32
4.2	Uživatelé databáze	32
4.2.1	Zaměstnanci a databáze	32
4.2.2	Zákazníci a databáze	33
4.3	Konceptuální návrh	33
4.3.1	Identifikace entit	33
4.3.2	Pololetní období.....	36
4.4	Logický a fyzický návrh.....	39
4.4.1	Tabulky atributů a číselníky	39
4.4.2	Evidence zaměstnanců	40
4.4.3	Evidence studentů a jejich zástupců	43
4.4.4	Zkouška.....	47
4.4.5	Hodnocení.....	48

4.5	Přínos návrhu	50
5	Závěr	52
	Seznam použitých zdrojů.....	53
	Monografické zdroje.....	53
	Internetové zdroje	54
	Seznam obrázků a tabulek	55
	Seznam obrázků.....	55
	Seznam tabulek.....	56
	Seznam příloh	57

Úvod

V současné době elektronické databáze neustále nahrazují papírové kartotéky nebo jiné zastaralé úložiště dat. Jedním ze zastaralých systémů ukládání dat se můžeme setkat ve školství, které by z důvodu velkého množství uchovávaných informací nemělo být výjimkou v postupu vpřed.

Přechod na čistě elektronické databáze by měl ulehčit práci jak učitelům, tak zvýšit komfort a možnosti studentům. Bohužel tento vývoj je brzděn nezkušeností, neznalostí ze strany učitelů a vysokými náklady, které si plno škol nemůže z důvodu nízkých rozpočtů dovolit. Společnost, se kterou spolupracuji, je Moravské gymnázium Brno s.r.o., již z názvu je poznat, že se jedná o soukromé gymnázium. Z důvodu vysoké konkurence na trhu musí firma nabízet lepší provedení služeb než ostatní. Silnou stránkou Moravského gymnázia v tomto směru jsou moderní technologie, kterými jsou prostory školy vybaveny. Tyto technologie otevírají spousty nových možností pro předávání vyučovaných vědomostí a zapojení samotných studentů do výuky.

1 Vymezení problému a cíl práce

Moravské gymnázium se snaží připravovat svoje studenty na způsob hodnocení vysokých škol už od prvního ročníku středoškolského stupně. Tato příprava je realizována pomocí přednášek a závěrečných pololetních zkoušek z vybraných předmětů. Termín těchto zkoušek byl volen po osobních konzultacích vyučujícího s žáky na hodinách předmětu, které byly tímto neefektivně využity. Dalšími nevýhodami je zápis pouze papírový u vyučujícího bez možnosti náhledu a s komplikovanou možnou změnou.

Řešením a zároveň cílem mojí práce bude návrh databáze, která bude páteří pro systém online registrování žáků na zkoušky. Součástí této databáze bude i evidence žáků a zaměstnanců školy, která je potřebnou inovací a nutností pro zdárné dosažení cíle. Databázi budu navrhovat v jazyce SQL a bude muset splňovat požadavky ředitele, učitelů a jednatele školy. Žáci by se zde měli seznamovat s výsledky zkoušky, které budou dostupné i pro rodiče žáků.

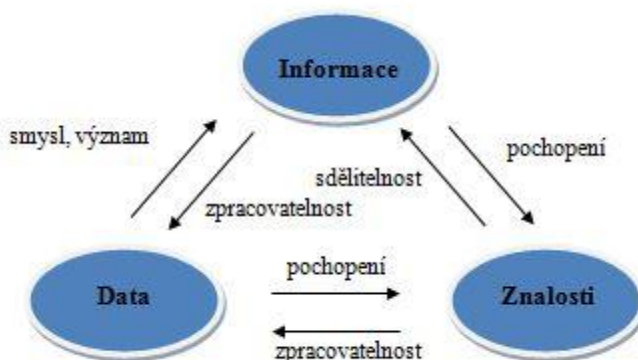
2 Teoretická východiska

V této části práce se seznámíme se základními teoretickými podklady, bez kterých by nebylo možné dosáhnout zvolených cílů. Nejprve si charakterizujeme základní pojmy a pojem databáze obecně. Následně budeme směřovat k samostatným krokům tvorby databáze, které se musí uskutečnit. Na konec zmíním jazyk SQL, ve kterém budu zadanou databázi tvořit.

2.1 Data, informace a znalosti

Data lze charakterizovat jako záznam určený k zpracování a přenosu. K těmto účelům jsou využívány různé technologie na rozdílných médiích (zápis na papíře, libovolná posloupnost znaků, hlasový záznam). Samy o sobě pro nás data nemusí nic znamenat, sdělovat a být jen součástí záznamu nebo posloupnosti. Za podmínky, že prohlízející rozumí významu zaznamenaných dat, jsou pro něj informací. Informacemi jsou tedy nazývány až data s konkrétním významem. Dělení informací je možné podle mnoha hledisek (6).

Po zjištění dat a informací získáváme určitou znalost, kterou můžeme aplikovat při uskutečnění určité události. Jestliže obdržíme informaci o poškození vozidla, na základě znalostí vyvodíme nutné činnosti – zastavení vozidla, kontrola vozidla, zavolání technika a následná oprava (6).



Obrázek 1 – Data, informace a znalosti
zdroj: vlastní zpracování

2.2 Databáze

Databází je bráno místo na kterém jsou uchovávána data o vybrané skupině osob, produktů, služeb a dalších součástí reálného světa, abychom je mohli snadno přidávat, upravovat a mazat. Primitivním příkladem databáze je zápisník učitele, který si zaznamenává termíny zkoušek pro každého z žáků a jeho hodnocení. Zápisník je rozdělen do různých skupin podle žákovi třídy. Toto je však databáze papírová, kterou se budeme snažit nahradit elektronickou formou (7).

Databáze jsou složené z entit, které popisujeme pomocí atributů a spojujeme je pomocí vzájemných vazeb mezi nimi. Základní modely jsou určeny podle způsobu ukládání dat a druhu vazeb vytvořených mezi nimi (7).

2.2.1 Lineární datový model

Mezi jednotlivými skupinami dat nejsou stanoveny jejich vazby, není tedy možné určit vzájemný vztah. Budeme mít tabulku s informacemi o studentech (RČ, jméno, třídu), tabulku s údaji o zkoušce (datum, počet míst) a tabulku učitelů (RČ, jméno). V lineárním modelu nejsme schopni určit vazby mezi jednotlivými tabulkami. Z toho plyne, že nemůžeme správně zjistit kdy, který student půjde na zkoušku a kterou ze zkoušek vypsal vybraný učitel. Z důvodu, že každá tabulka je brána, jako samostatný objekt nám tento datový model nevyhovuje (1).

2.2.2 Hierarchický datový model

Na rozdíl od lineárního modelu můžeme vytvořit vazby mezi jednotlivými tabulkami. Jedná se o víceúrovňový model, který můžeme promítnout ve stromové struktuře, kde nadřazená tabulka může mít více podřízených tabulek, ale obráceně tomu být nemůže. V našem případě by byla nadřazenou (rodičovskou) tabulka zkoušek, které by se podřizovaly na příklad tabulky studentů a předmětů. Z důvodu společného rodičovského segmentu bychom nemohli vytvořit vazbu mezi těmito dvěma dceřinými tabulkami. Z tohoto důvodu je model pro naši problematiku nevhodný, i když má

velkou výhodou v přehlednosti a rychlosti vyhledávání dat za pomoci vazeb mezi tabulkami (1).

2.2.3 Síťový datový model

Velmi podobným modelem je síťový datový model, který je v podstatě nástupcem hierarchického. Na rozdíl od něj zde nerozlišujeme rodičovské a dceřiné tabulky. Dalším rozdílem je způsob přístupu uživatele k datům. V tomto modelu může uživatel databáze přistupovat k datům z libovolného segmentu, aniž by musel přes kořenovou tabulku. Při implementaci modelu na předešlou problematiku zjistíme, že již můžeme vytvořit vazbu mezi žákem a předmětem, která nám umožní navíc vyhledat vztahy mezi jednotlivými žáky a různými předměty (1).

2.2.4 Objektový datový model

Nejmladším z datových modelů je objektový. Základním stavebním kamenem je objekt, který má kromě svých atributů definované i metody určující jeho chování v databázi. Máme-li vytvořený objekt student a jeho atributy (RČ, jméno, třídu), můžeme mít definovány metody (přidej studenta – proběhne kontrola možného duplicitního záznamu). Díky unikátnímu identifikátoru, který má každý objekt v databázi můžeme tvořit vazby podobně jako v síťovém modelu, které jsou rozšířené o možnost existence vazeb relačních (1).

2.2.5 Relační datový model

Relační model je řešením mnohých omezení, která nastávají u hierarchického a síťového modelu. Data relačního modelu jsou prezentována ve formě dvourozměrných tabulek. Jednotlivé věty tabulek můžeme identifikovat unikátním primárním klíčem, díky kterému můžeme definovat vzájemné vztahy a možné operace mezi nimi. Tato metoda je založena na ukládání velkého množství dat a práci s nimi. Relační model je stále nejvíce rozšířenou a používanou metodou, kterou jsem zvolil pro naši problematiku (1).

2.3 Terminologie

2.3.1 Entita

Entita reprezentuje určitou skupinu objektů existující v reálném světě, která se nazývá instancí. Všechny instance dané entity se označují populací a mají shodnou vnitřní datovou strukturu, která je vyjádřena množinou atributů. Pouhá shoda atributů ale samozřejmě nestačí, všechny instance dané entity musí mít i logickou souvislost. Entitou nemusí být jen nějaké reálné fyzické objekty (učitel, žák), ale může se jednat i o objekty abstraktní (vyučovaný předmět, zkouška). Každá entita má svůj vlastní název a charakterizující atributy. Každý atribut má vlastní datový typ. Jde o základní datové typy, se kterými se setkáme ve většině databázových systémů (5).

2.3.2 Vazby mezi entitami

Vazba mezi entitami představuje vzájemný logický vztah mezi nimi. Jsou rozděleny na tři základní typy podle kardinality. Kardinalita je jev, který určuje počet subjektů vstupujících do vztahu (6).

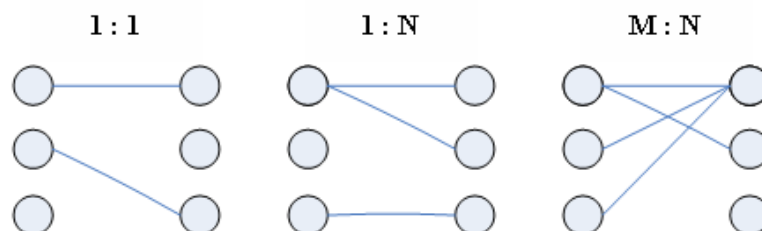
1:1 – Jednomu záznamu v jedné tabulce, odpovídá právě jeden záznam v tabulce druhé.
Příklad: Učitel vyučuje právě jeden z předmětů (6).

1:N – Jednomu záznamu v jedné tabulce, odpovídá více záznamů v tabulce druhé.

Příklad: Učitel vyučuje alespoň dva různé předměty (6).

N:M – Více záznamů v jedné tabulce, odpovídá více záznamům v tabulce druhé.

Příklad: Učitel vyučuje více předmětů, jeden z nich vyučují alespoň dva učitelé (6).



Obrázek 2 - Vazby mezi entitami, převzato z (6)

2.3.3 Klíče tabulek

V relačních databázích se určují tři druhy klíčů, které lze za splnění podmínky jedinečnosti přiřadit ke každému záznamu (6).

Kandidátní klíč – je atribut, nebo kombinace atributů, které jednoznačně identifikuje každý záznam v tabulce. Nemůže obsahovat NULL hodnotu a musí být jedinečný (6).

Primární klíč – je vybraný kandidátní klíč k identifikování jedinečných záznamů v tabulce pro celou databázi (6).

Cizí klíč – cizí klíč odkazuje na svou podobu primárního klíče v jiné tabulce. Tímto odkazem vzniká relace mezi tabulkami (6).

2.3.4 Relační integrita

Pro relační databáze existuje několik druhů omezení, která aplikujeme na jednotlivé tabulky a sloupce v nich (6).

Entitní integrita – pro každou tabulku musíme určit primární klíč, který nesmí nabývat hodnoty NULL (6).

Omezení domén – omezení dat, které je možné vkládat do daného sloupce (6).

Referenční integrita – zajišťuje vytvoření záznamu pouze s cizím klíčem, ke kterému existuje totožný primární klíč v jiné tabulce, jinak musí cizí klíč nabývat hodnoty NULL (6).

2.3.5 Normalizace

Normalizace je základním principem relačního modelu. Je to proces pro zpracování dat, které chceme v databázi uchovávat. Díky tomuto procesu předcházíme vzniku nadbytečných dat a zachováváme jejich integritu. Relační model původně obsahoval tři normální formy, i když později byly přidány další. Pro pokrytí téměř všech nedostatků databáze by měly stačit původní tři normální formy, které by měl návrh splňovat (1).

1. normální forma

První normální forma je splněna za podmínky, že všechny atributy jsou dále nedělitelné. Tato forma bývá splněna i za předpokladu, že jednotlivé části údaj nepotřebujeme a rozhodneme se je zanechat složené např. bydliště uživatele (1).

2. normální forma

Podmínkou pro splnění druhé normální formy je platná první normální forma a hodnoty každého ze sloupců, které tvoří primární klíč musejí být plně závislé na hodnotách tvořící primární klíč. To znamená, že nesmí existovat záznam, který by byl závislý jen na části primárního klíče (1).

3. normální forma

Pokud je relace v druhé normální formě a současně nejsou atributy vzájemně závislé, tak se nachází relace ve třetí normální formě.

Variací třetí normální formy je Boyce – Coddova normální forma, ve které je stav relace shodný s třetí normální formou a dále platí, že mezi kandidátními klíči není žádná funkční závislost (1).

4. normální forma

Relace je ve čtvrté normální formě pokud splňuje podmínky Boyce – Coddovi normální formy a současně jsou všechny vícehodnotové závislosti funkčními. To znamená, že čtvrtá normální forma vyžaduje, aby klíč tvořily jen hodnoty se vzájemnou souvislostí (1).

5. normální forma

Relace je v páté normální formě za podmínky, že je ve čtvrté normální formě a není možné přidat nový atribut, aniž by vlivem skrytých závislostí nedošlo k rozpadu na několik dílčích tabulek (1).

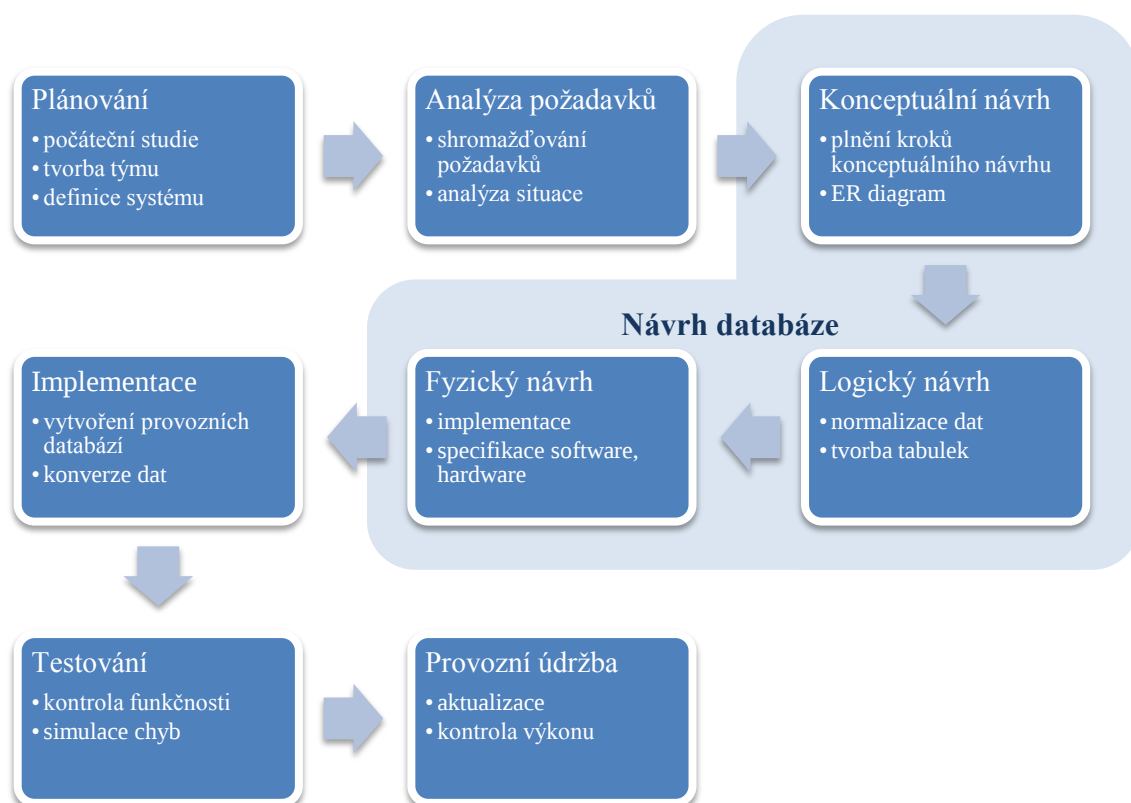
2.4 Tvorba návrhu databáze

Databáze obsahují mnoho druhů rozdílných informací. Z tohoto důvodu se stává návrh obtížným. Špatný návrh může způsobovat neefektivní vyhledávání, nebo úplnou

nedostupnost potřebných informací. V následujících stranách si popíšeme správný postup při návrhu databáze.

2.4.1 Životní cyklus databáze

V okamžiku potřeby databáze začíná její životní cyklus, který je zakončen s odstavením databáze. Při dlouhodobé práci se cyklus rozděluje do několika projektů, které mají vlastní cíle, které dohromady tvoří kompletní databázi (1).



Obrázek 3 – Životní cyklus databáze
zdroj: vlastní zpracování

2.4.2 Konceptuální návrh

Návrh je sestaven na základě dat používaných uživateli. Cílem konceptuálního návrhu je vytvoření ER modelu, který bude splňovat požadavky na databázi. Samotný postup

návrhu je rozdělen do devíti kroků, které postupně upřesňují finální podobu ER modelu (1).

Identifikace entit - Cílem tohoto kroku je definice hlavních objektů databáze – entit. Získány jsou z požadavků uživatele na objekty a data. Výsledkem je datový slovník, který obsahuje údaje o názvu entity, její popis, předpokládaný počet výskytů a případně alias entity (1).

Identifikace relací – V tomto kroku se určují vztahy mezi entitami databáze. Přesnější vyjádření relací se získá po zanesení modelu multiplicity, který se používá ke kontrole a kvalitě dat. Výsledek může být prvotní ER diagram nebo datový slovník, které obsahují názvy entit, vztah a multiplicitu (1).

Identifikace atributů – Jedná se o zjištění faktů, které se stanou atributy entit modelu. Atributy dělíme na jednoduché, složené, vícehodnotové a odvozené. Výsledkem tohoto kroku je přehled nejlépe ve formě tabulky s údaji charakterizující atribut – jméno, popis, typ, délka, aliasy, nutnost zadání, druh (1).

Určení domén atributů – Cílem je určení množiny hodnot, z nichž čerpají hodnoty jeden nebo více atributů (1).

Určení kandidátních a primárních klíčů – Probíhá výběr všech kandidátních klíčů v rámci entity, z kterých následně zvolíme primární klíč. Zbývající kandidátní klíče značíme jako alternativní (1).

Specializace/generalizace entit – Modelování podtříd a nadtříd entit, které je potřeba vzájemně odlišit. Tento krok je volitelný (1).

Kontrola redundance modelu – Přezkoumání všech vazeb v modelu, pokud se nachází vazba 1:1, tak při identifikaci entit došlo k přiřazení dvou ke stejnému objektu. Další činností je odstranění redundantních relací (1).

Kontrola uživatelských transakcí – Na základě požadavků uživatele na transakce kontrolujeme model, zda li umožňuje tyto transakce. Kontrola je prováděna popisem transakce nebo sledováním cest transakcí (1).

Diskuze nad konceptuálním návrhem s uživateli – Seznámení uživatele s výsledkem konceptuálního návrhu, jeho kontrola, návrhy na změny a případné úpravy (1).

2.4.3 Logický návrh

Návrh vychází z finální verze ER modelu, vytvořeném během konceptuální části. Cílem je vytvoření tabulek, jejich popis, kontrola struktury, integritních omezení a podpory transakcí. Průběh návrhu je rozdělen do pěti kroků (1).

Vytvoření tabulek – V tomto kroku vytváříme tabulky, které byly navrženy v konceptuálním modelu jako entity, relace a jejich atributy. Relace v ER diagramu jsou různého typu, proto upravujeme strukturu z důvodu určení rodičovských a dceřiných tabulek. Výsledkem je seznam tabulek, který je identifikován jejich názvem, seznamem atributů, datové typy, délky a definovanými klíči (1).

Kontrola struktury tabulek – Kontrolujeme vytvořené tabulky z prvního kroku, jestli jsou minimálně ve třetí normální formě. Pokud ne, musíme provést rekonstrukci modelu nebo tabulky, aby splňovali podmínky normalizace (1).

Kontrola uživatelských transakcí – Kontrola nyní probíhá z pohledu tabulek. Ujistíme se, zda model stále splňuje požadavky uživatele na transakce (1).

Kontrola integritních omezení – Zjišťujeme, zda logický návrh obsahuje všechna potřebná integritní omezení – NULL, omezení domén, referenční integrita a další (1).

Diskuze nad logickým návrhem s uživateli - Seznámení uživatele s výsledkem logického návrhu, jeho kontrola, návrhy na změny a případné úpravy (1).

2.4.4 Fyzický návrh

Převod logického návrhu do podoby systémového softwaru a na vnější paměťové zařízení a jeho následná specifikace. Návrh je rozdělen do šesti kroků (1).

Převod logického návrhu databáze do cílového DBMS – Jedná se o návrh tabulek pro konkrétní DBMS, reprezentace odvozených dat a zajištění ostatních integritních omezení (1).

Volba organizace souborů a indexů – Cílem je zlepšení výkonnosti databází. Za tímto účelem probíhá analýza transakcí a následný výběr organizace souborů. V případě potřeby se zavádí indexy. U menších databází je možné tento krok vynechat (1).

Návrh uživatelských pohledů – Navrhnutí požadovaných pohledů (propojení sloupců různých tabulek) podle potřeb uživatele (1).

Návrh bezpečnostních mechanismů – Cílem je zabezpečení dostupnosti dat jednotlivým uživatelům pomocí přihlašovacích údajů a nastavení pravomocí (1).

Zvážení kontrolovaného zavedení redundance – Zvažuje se, zda porušení některých pravidel normalizace neprospěje výkonnosti databáze. U menších databází se tento krok vynechává (1).

Monitorování a vyladění systému v provozu – Kontrola finální verze návrhu. Testování rychlosti dotazů, doby odezvy, ověření funkčnosti uživateli atd. (1).

2.5 SQL (Structured Query Language)

Zkratka SQL znamená strukturovaný dotazovací jazyk. Tento jazyk je standardizovaný a používán pro práci s daty relačních databází. Prvotní verze je známa z poloviny 70. let pod názvem SEQUEL od firmy IBM, která se zabývala výzkumem relačních databází. Jazyk SQL se skládá ze čtyř částí (3).

2.5.1 Základní dělení SQL

Jazyk pro definici dat

Tato skupina příkazů umožňuje tvorbu, mazání nebo úpravu částí databáze. Obsahuje například tyto příkazy (2):

- **CREATE DATABASE** – vytvoření databáze
- **CREATE TABLE** – vytvoření tabulky
- **DROP TABLE** – smazání tabulky
- **ALTER TABLE** – úprava tabulky

Jazyk pro manipulaci s daty

Tato skupina příkazů umožňuje vybírat, vkládat, mazat nebo upravovat data v již vytvořených tabulkách. Obsahuje například tyto příkazy (2):

- SELECT – výběr dat
- INSERT – vložení dat
- DELETE – smazání dat
- UPDATE – úprava dat

Jazyk pro správu práv

Těmito příkazy můžeme přidělovat nebo brát určitá práva pro uživatele databáze. Obsahuje například tyto příkazy (2):

- CREATE USER – vytvoření uživatele
- DROP USER – smazání uživatele
- ALTER USER – úprava uživatele
- GRANT – přidělení privilegia

Jazyk pro řízení transakcí

Těmito příkazy řídíme transakce probíhající v databázi. Obsahuje například tyto příkazy (2):

- START TRANSACTION – zahajuje transakci
- COMMIT – potvrzuje a ukončuje transakci
- ROLLBACK – odvolává zahájenou transakci

2.5.2 Datové typy v SQL

Numerické typy (4):

INTEGER – celá čísla (rozsah od -2147483648 do +2147483647)

SMALLINT – malá celá čísla (rozsah od -32768 do +32767)

TINYINT – malá celá čísla (rozsah od 0 do 255)

NUMERIC (p,s) – desetinná čísla s celkem p čísly a s desetinou čárkou s čísel zprava

FLOAT (p) – reálná čísla s plovoucí desetinnou čárkou, s p platnými číslicemi, max. 38

REAL (p) – reálná čísla s plovoucí desetinnou čárkou, s p platnými číslicemi, max. 18

Znakové řetězce (4):

CHARACTER (n) řetězec znaků o délce n , max. 8 000, má-li řetězec méně znaků než n , je doplněn zprava prázdnými znaky

CHARACTER VARYING (n) řetězec znaků o délce n , max. 8 000, řetězec kratší než n se nedoplňuje prázdnými znaky

TEXT (max) – jako CHARACTER VARYING, max. 8 000 bytů

Datum a čas (4):

DATE – datum ve tvaru *rrrr-mm-dd*

TIME – čas ve tvaru *hh:mm:ss*

SMALLDATETIME – čas ve tvaru *rrrr-mm-dd hh:mm:ss*

TIMESTAMP – struktura obsahující datum a čas

3 Analýza problému a současného stavu

V této kapitole seznámím s danou problematikou, situací a samotnou spolupracující společností (školou), která požaduje vytvoření databáze. Dále si charakterizujeme hardwarové a softwarové vybavení budovy gymnázia. Nakonec se seznámíme s nejrozšířenějšími školními informačními systémy dostupnými na trhu.

3.1 Základní údaje společnosti

Název: Moravské gymnázium Brno s.r.o.

Sídlo: Veveří 30, 60200 Brno

Právní forma podnikání: Společnost s ručením omezeným

IČ: 6348997

Datum zápisu: 18. 12. 1995

Společnost byla založena za účelem provozování soukromé školy. Moravské gymnázium má dlouholetou tradici a je nejstarším soukromým gymnáziem v Brně. Budova školy se nachází v centru města Brna na ulici Veveří, což je značnou výhodou pro studenty i zaměstnance školy, kteří bydlí mimo město Brno a dojíždí z blízkého okolí. Moravské gymnázium každoročně dosahuje vysokého procenta přijatých studentů na vysoké školy (88,6% v roce 2011) a úspěšných výsledků studentů u mezinárodních jazykových zkoušek. Dále získalo velmi dobré hodnocení od České školní inspekce, je zařazeno mezi přidružené školy UNESCO a obdrželo zlatý certifikát od společnosti SCIO.

Moravské gymnázium nabízí čtyřleté a osmileté studium s profilací (humanitní nebo anglické), mezinárodní zkoušky z anglického, německého, francouzského a španělského jazyka, studijní pobyty v zahraničí, besedy s úspěšnými lidmi z různých oblastí, poznávací zájezdy do zahraničí (Tokio, New York, Londýn), kvalifikovaný

pedagogický sbor. Studenti se účastní literárních, výtvarných, sportovních a matematických soutěží.

Mezi nadstandardní nabídku, kterou se snaží Moravské gymnázium předčít státní školy, patří zvýšená dotace hodin cizích jazyků, které jsou řešeny konverzacemi s rodilými mluvčími. Individuální přístup ke studentům s poruchami učení a individuální studijní plán jsou samozřejmostí. Gymnázium pořádá přípravné semináře k přijímacímu řízení na vysoké školy. Rodiče studentů mají možnost navštívit výuku a vznést připomínky, návrhy na její provedení, které jsou řešeny vedením školy. Dlouholetou tradicí je spolupráce se zahraničními školami z Irska, Španělska, Francie, Itálie, Rakouska a Slovenska.

V čele společnosti Moravské gymnázium Brno s.r.o. je jednatel Ing. Jaromír Jeřábek. O vedení školy se po finanční stránce stará ekonomka a po školní stránce ředitel, kteří se zodpovídají přímo jednatele, který má hlavní a poslední slovo ve všech případech. Poslední součástí hierarchie jsou administrativní pracovníci a učitelé.



Obrázek 4 – Organizační hierarchie
zdroj: vlastní zpracování

3.2 Analýza současné situace

V současné době jsou údaje o studentech uchovávány v papírových třídních výkazech a ve starých databázích. Třídní výkazy jsou zakládány současně se vznikem nové třídy. Výkaz se skládá z obálky, na které jsou základní identifikační údaje spojující žáky. Vnitřní obsah výkazu tvoří dvoustránky věnovány jednotlivým studentům. Na každé dvoustránce jsou zaznamenány osobní údaje studenta, hodnocení, informace o docházce a další změny v průběhu studia. Škola musí tyto údaje uchovávat i několik let po skončení studia, z tohoto důvodu jsou staré výkazy skenovány do elektronické podoby pro lepší uchování dat.

Jednou z odlišností od ostatních středních škol je příprava studentů na vysokoškolský způsob výuky. Studenti již od prvních ročníků středoškolského vzdělání mají společné hodiny vybraných předmětů, které jsou uskutečňeny v největší učebně na způsob přednášky. Přednášená látka je následně probírána v každé třídě zvlášť. Celkové hodnocení předmětu je z hlavní části pololetní zkouškou, která je uskutečněna na konci pololetí. Zkoušky probíhají ve většině případů v ústní formě během jednoho týdne a jsou rozděleny do několika termínů po určitém počtu studentů. Tento způsob zkoušení umožňuje lepší a spravedlivější kontrolu znalostí studentů. Přidělení termínů probíhá během vyučující hodiny, kde se studenti zapisují na papír k vybranému dni a času. Zápis touto formou je neefektivní kvůli ztrátě času a nepraktický z důvodu vyžadování změny a přehledu nad termíny. V případě změny musí student za učitelem a přepsat termín osobně. Vysoká četnost výskytu tohoto případu vznáší do zápisu na papír zmatek a nepřehlednost. Studenti nepřítomní ve výuce jsou znevýhodněni, protože nemají možnost volby. Výsledky se studenti dozvědí v případě ústní zkoušky okamžitě, nebo na internetu, kde jsou hodnocení zkoušky zveřejněna.

3.3 Informační technologie gymnázia

Moravské gymnázium využívá nejmodernější technologie pro uspokojení požadavků klientů a k usnadnění práce zaměstnancům. Výuka je díky těmto moderním pomůckám názornější a efektivnější.

3.3.1 Hardwarové vybavení

V každé třídě jsou umístěny moderní interaktivní dataprojektory Epson EB-450Wi, které nahradily kombinaci interaktivní tabule a staršího dataprojektoru značky Toshiba. Tyto dataprojektory jsou navrženy tak, aby vytvořily dokonalý obraz ve všech učebnách o různých velikostech. Dataprojektory jsou připevněny nad keramickou tabulí, což umožňuje technologie zrcadlového zobrazování. Díky této technologii je promítání bez vytváření stínů na obrazovce, či oslňování a umožňuje neomezený pohyb v prostoru tabule, který je nezbytnou součástí skoro každé výuky. Technologie 3LCD, která je součástí dataprojektoru zaručuje kvalitu promítání za jakýchkoli světelných podmínek, včetně denního světla. V každé třídě jsou dataprojektory připojeny k průměrně výkonnému počítači, který slouží pro výuku a o přestávkách je dostupný studentům, kteří většinou využívají k poslechu hudby.

Počítačová učebna je vybavená nejmodernějšími počítačovými sestavami, které jsou pravidelně obměňovány. Učebna je mimo výuku dostupná studentům gymnázia. Sborovny a sekretariát jsou vybaveny počítači, které v minulosti sloužily jako výukové v počítačové učebně. Nejméně výkonný počítač je od ministerstva školství, který slouží ke státním maturitám a je neustále zapnutý. Součástí vybavy sboroven a sekretariátu je několik kopírek, tiskáren a skenerů různých značek. Za zmínku stojí pouze kopírka Océ 3165, která je dostupná pro studenty i zaměstnance. Její výkon a funkce jsou dostačující pro potřeby uživatelů.

Jako technická místnost v budově slouží vrátnice. V místnosti je racková skříň značky Eurocase Rack Cabinet, ve které je hlavní připojení k internetu a umístěny server, externí velkokapacitní harddisk a několik dvaceti portových switchů. V každém patře budovy je jeden osmi portový switch a WiFi router.

Nejnovější využívanou technologií na gymnáziu je čtečka otisků prstu, která je umístěna, z důvodu bezpečnosti a vytížení školníka, na vstupních dveřích školy.

Hardware	
Druh	Počet
Dataprojektory	16
PC	52
Tiskárny	9
Kopírky	3
Skenery	4

Tabulka 1 – Přehled hardwaru
zdroj: vlastní zpracování

3.3.2 Softwarové vybavení

Veškeré počítačové sestavy v budově mají operační systém Windows 7, který je legálně zakoupen. Jako kancelářský balík je využívána verze Microsoft Office 2007, ke které jsou zakoupeny licence. Nejnovější software je k čtečce otisků prstů, který kontroluje totožnost studentů a zaměstnanců školy, v databázi jsou uloženy otisky obou ukazováčků, které byly vzaty na začátku školního roku. Gymnázium vlastní i spoustu jednoduchého výukového softwaru, který využívají učitelé během svých hodin k zlepšení kvality výuky.

3.4 Školní informační systémy

V současné době jsou školní informační systémy nepostradatelnou součástí stále více středních škol. Systémy jsou ve většině případů složeny z několika modulů, které zahrnují v podstatě celou administrativu škol. Výběr vhodného informačního systému bývá ovlivněn podle několik kritérií:

- výrobce – společnost, která dodává školní IS již několik let, bude pravděpodobněji pokračovat než společnost nově založená.
- podpora, servis – většina dodavatelů nabízí odborná školení, která jsou potřebné k správnému užívání. V případě nedostatků, nebo chyb se můžeme setkat s existující konzultační linkou, seznamem často kladených otázek nebo technickou podporou servisních pracovníků. Součástí podpory by také měly být aktualizace, které se dnes běžně uskutečňují pomocí internetu.

- rozsah – některé IS na trhu mohou zahrnovat pouze několik částí školní administrativy a nabízet je jako jednotlivé části. V případě plánování postupného rozšiřování IS nebo změny potřeb školy je důležité, aby existovaly veškeré požadované moduly ke správné implementaci.
- cena – z pravidla je určována rozsahem IS, podle počtu modulů, které obsahuje a podle velikosti ukládaných dat a školy. Pořizovací cena by rozhodně neměla být hlavním kritériem při volbě IS.
- informovanost – kvalitní webová prezentace s možnými náhledy a demo verzemi by neměla chybět u žádného prodejce IS.

3.4.1 SAS

Je softwarový informační systém nabízený společností MP-Soft a.s., která se na trhu pohybuje již přes 15 let. Od založení se společnost zabývala vývojem odborných výukových programů a elektronických příruček, stala se i hlavním sponzorem veletrhu vzdělávání GAUDEAMUS. Softwarový balík je určený pro základní, střední a vyšší odborné školy.

Náplň SAS:

- vedení školní matriky
- evidence žáků, jejich klasifikace, přijímací řízení
- evidence pracovníků školy, majetku, knihovny
- rozvrh hodin s automatickým nasazováním lístků, suplování
- tisk seznamů, vysvědčení, rozvrhů

Užívání systému SAS je řízeno licenční politikou, podle skutečného počtu uživatelů. Ceník služeb je přehledně dostupný na webových stránkách společnosti, která nabízí věrnostní programy a technickou podporu pro zákazníky (9).

3.4.2 Škola OnLine

Škola OnLine je manažerský interaktivní IS, jehož součástí jsou hostované aplikace Katedra, Žákovská, Akademie, Spisovka a OLAT. Nad tímto systémem převzalo záštitu Ministerstvo školství. Aplikace Katedra, Žákovská a Spisovka jsou určené mateřským,

základním, středním a vyšším odborným školám. Katedra a Spisovka jsou aplikace pro učitele a školu, kdežto Žákovská je určena pro žáky/studenty a rodiče. K používání aplikace stačí pouze internetové připojení, tudíž škola nemusí nic instalovat.

Náplň Katedry:

- vedení školní matriky
- klasifikace žáků, přijímací řízení
- rozvrhy hodin, suplování, osobní kalendář uživatelů
- evidence majetku, knihovny
- tisk seznamů, vysvědčení, rozvrhů
- e-learningová výuka

Náplň Žákovské:

- docházka, studijní výsledky
- osobní kalendář, rozvrhy
- učební materiály, knihovna
- plány zkoušení

Cena se pro každou školu stanovuje na základě počtu žáků/studentů a vybraných modulů. Ceník služeb je přehledně dostupný na webových stránkách, kde je možné objednat veškerá školení týkající se nabízené aplikace (10).

3.4.3 Bakaláři

Je IS nabízený Bakaláři Software s.r.o., kteří se věnují pouze této problematice. Bakaláři umožňují volitelně uložit data na SQL serveru. Podporován je Microsoft SQL server, k provozu postačují verze zdarma. Díky otevřenosti a variabilitě vyhoví systém mateřským, základním, středním a vyšším odborným školám. Produkt je vytvořený přímo na problematiku školství v ČR.

Náplň Bakalářů:

- evidence žáků, jejich klasifikace, přijímací řízení
- evidence pracovníků školy, majetku, knihovny
- rozvrh hodin s automatickým nasazováním lístků, suplování

- tisk seznamů, vysvědčení, rozvrhů

Ceník služeb je přehledně dostupný na webových stránkách, kde se cena počítá podle počtu vybraných modulů a počtu žáků/studentů kteří budou evidováni. Možné je objednání školení, která mohou být na zákazníkem požadované téma přímo u něj (8).

3.4.4 Souhrnný přehled

Vybrané IS mají hodně podobná kritéria. Všechny nabízejí pokrytí širokého okruhu školní administrativy, informovanost je na vysoké úrovni a školení jsou standardem. Rozdílem mezi IS je cena, jejich vzhled a způsob provedení. Žádný z informačních systémů neobsahuje modul pro výpis a následnou online registraci zkoušek studentů.

	SAS	Škola OnLine	Bakaláři
výrobce	15 let tradice	podpora ministerstva školství	specifické zaměření
podpora	počáteční školení zdarma	placené školení	široké možnosti školení
rozsah	všechny moduly	všechny moduly	všechny moduly
cena	licenční politika, podle skutečného počtu uživatelů, velikosti školy	licenční politika, podle skutečného počtu uživatelů, velikosti školy	licenční politika, podle skutečného počtu uživatelů, velikosti školy
informovanost	náhledy, videoukázky	náhledy, zkouška zdarma	náhledy, videoukázky

Tabulka 2 – Přehled IS
zdroj: vlastní zpracování

4 Návrh vlastního řešení, přínos návrhu

V této kapitole budu navrhovat možné řešení k vznešeným požadavkům na databázi pro zadavatelské gymnázium. První fází návrhu bude konceptuální část, během které se postupně dopracujeme k ER diagramu. Při návrhu logického a fyzického schéma budu postupovat podle jednotlivých částí samotné databáze. U každé z částí charakterizují sestavy tabulek, z kterých se skládají, a pokusím se navrhnout možné procedury.

4.1 Požadavky na databázi

Požadavky na data, které budou ukládány v databázi, byly několikrát konzultovány s jednatelem a ředitelem školy. Prostředí a způsob fungování gymnázia znám dobře díky několik let dlouhé spolupráci, což velice usnadnilo průběh konzultací. Hlavním požadavkem bylo vytvoření databáze, která plnohodnotně nahradí papírové třídní výkazy. Databáze by měla být současně základním pilířem pro plánovaný online systém registrace zkoušek, který by měl nahradit současný způsob. Velká důležitost je kladena na možnost rozšíření a úpravy databáze k potřebám školy. V budoucnu by se jednalo o rozšíření typu: tvorba rozvrhů, elektronické třídnice a dalších funkcí, které by nahradily a usnadnily běžné školní aktivity.

4.2 Uživatelé databáze

K databázi budou přistupovat dvě hlavní skupiny uživatelů. Jednou ze skupin budou zaměstnanci, tedy učitelé. Druhou skupinou budou rodiče a studenti školy, tedy zákazníci.

4.2.1 Zaměstnanci a databáze

Na gymnáziu je v současnosti třiatdvacet učitelů, ředitel a jednatel, kteří budou mít přístup do databáze. Všichni učitelé jsou schopní pracovat s počítačem a vykonávat potřebné činnosti, které s databází budou souviset. Změna je vítaná, protože zapisování

údajů do papírových výkazů je nepřehledné a nepohodlné z důvodu vysokého počtu záznamů a není možná manipulace více uživatelů současně. Vypsání termínu nevyžaduje čekání na zpětnou reakci studentů, tudíž celý proces bude časově méně náročný. V případě počátečních problémů nebo neznalosti budou mít k dispozici učitele informatiky a výpočetní techniky. Tato skupina uživatelů bude s databází provádět operace na vytvoření záznamů o studentech, termínů zkoušek a jejich následné vyhodnocení.

4.2.2 Zákazníci a databáze

Zákazníky je brána skupina tvořená rodiči a studenty gymnázia. Pro tuto skupinu je inovace formou online registrace vítanou a vyžadovanou. Rodiče uvítají lepší kontrolu prospěchu svých dětí, která se často vyskytuje s potížemi. Studenti kvůli pohodlnějšímu způsobu registrace a možnosti změny termínu, které nebudou vyžadovat osobní styk s učitelem a budou mít možnost vykonat tyto aktivity v pohodlí svého domova i v nepříznivém zdravotním stavu. Tato skupina uživatelů bude využívat databázi za cílem se registrovat na termíny zkoušek a prohlížet svoje výsledky.

4.3 Konceptuální návrh

Cílem návrhu je vytvoření ER diagramu na základě dat používaných uživateli. Tvorba diagramu je rozdělena do několika kroků, které identifikují entity a relace v databázi. Zakončení návrhu probíhá posouzením s uživateli.

4.3.1 Identifikace entit

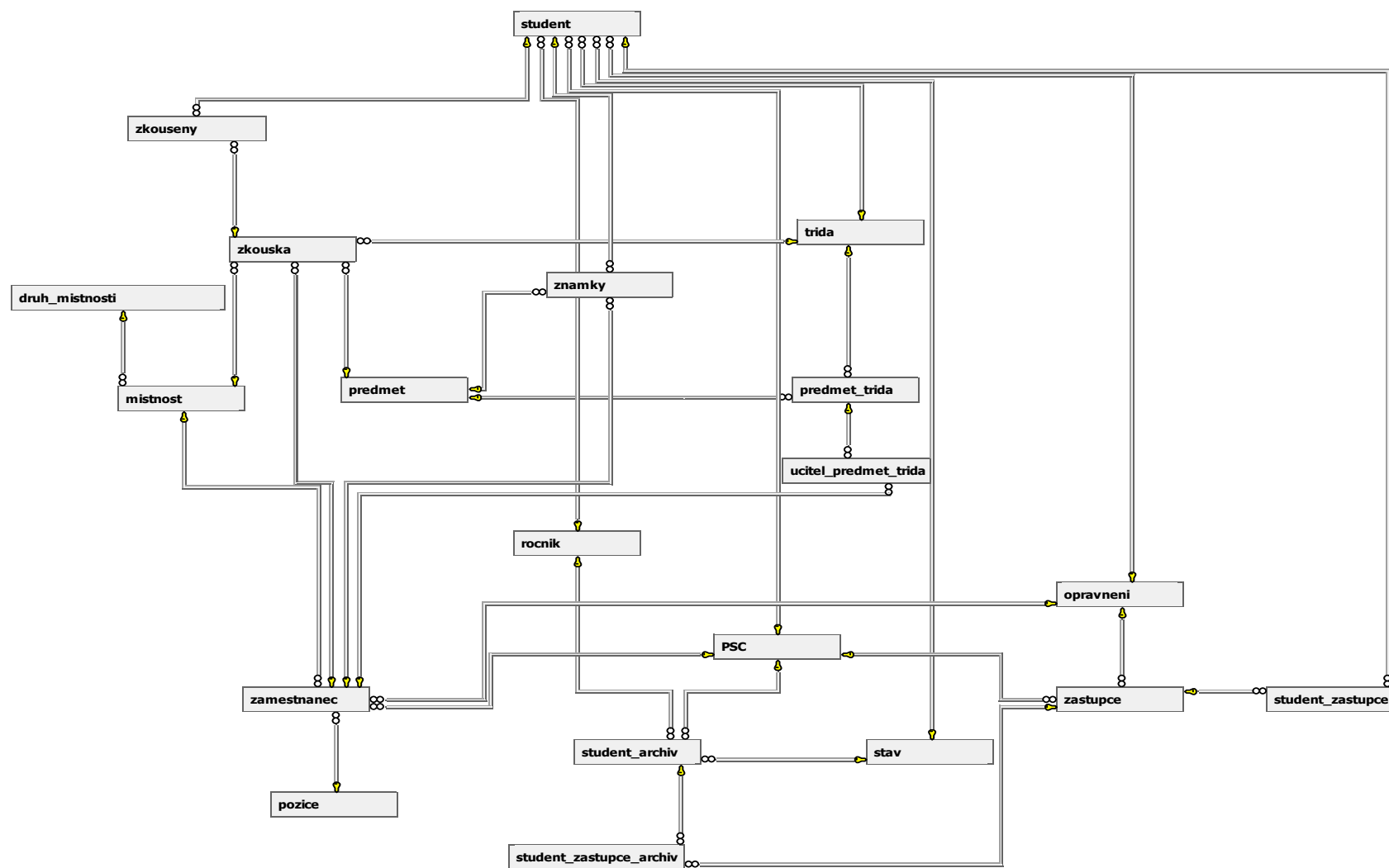
V tomto kroku identifikujeme hlavní objekty databáze – entity. Výsledkem je následující tabulka, která zobrazuje entity, jejich popis a počet předpokládaného výskytu.

Název entity	Popis entity	Počet výskytů
druh_mistnosti	druhy místností v budově	10
mistnost	seznam místností v budově	40
opraveni	oprávnění uživatele	5
pozice	pozice zaměstnance školy	9
predmet	předměty vyučované na škole	20
predmet_trida	úrovně jednotlivých předmětů	asi 150
PSC	PSČ jednotlivých měst	50
rocnik	ročník studia, jeho délka	30
stav	stav studia žáka	4
student	seznam studentů	asi 320
student_archiv	archiv studentů	asi 500
student_zastupce	přiřazení zástupců k aktivním studentům	asi 600
student_zastupce_archiv	přiřazení zástupců k archivovaným studentům	asi 1000
trida	seznam tříd na gymnáziu	12
ucitel_predmet_trida	předměty, které vyučují konkrétní učitelé	asi 300
zamestnanec	seznam zaměstnanců gymnázia	31
zastupce	seznam zákonných zástupců studentů	asi 1500
zkouseny	přihlášení studenti na zkoušku	asi 1800
zkouska	pololetní zkouška	asi 600
znamky	bodové hodnocení studentů	asi 15000

Tabulka 3 – Identifikace entit
zdroj: vlastní zpracování

Výsledky dalších kroků konceptuálního návrhu jsou vloženy v přílohách. Příloha číslo jedna je kompletní datový slovník tvořené databáze. V příloze číslo dva je zobrazen ER diagram s veškerými atributy a klíči tabulek.

Na následující stránce je vyobrazen finální ER diagram pouze s názvy tabulek, podle kterého budu v následujících krocích tvořit logický a fyzický návrh.

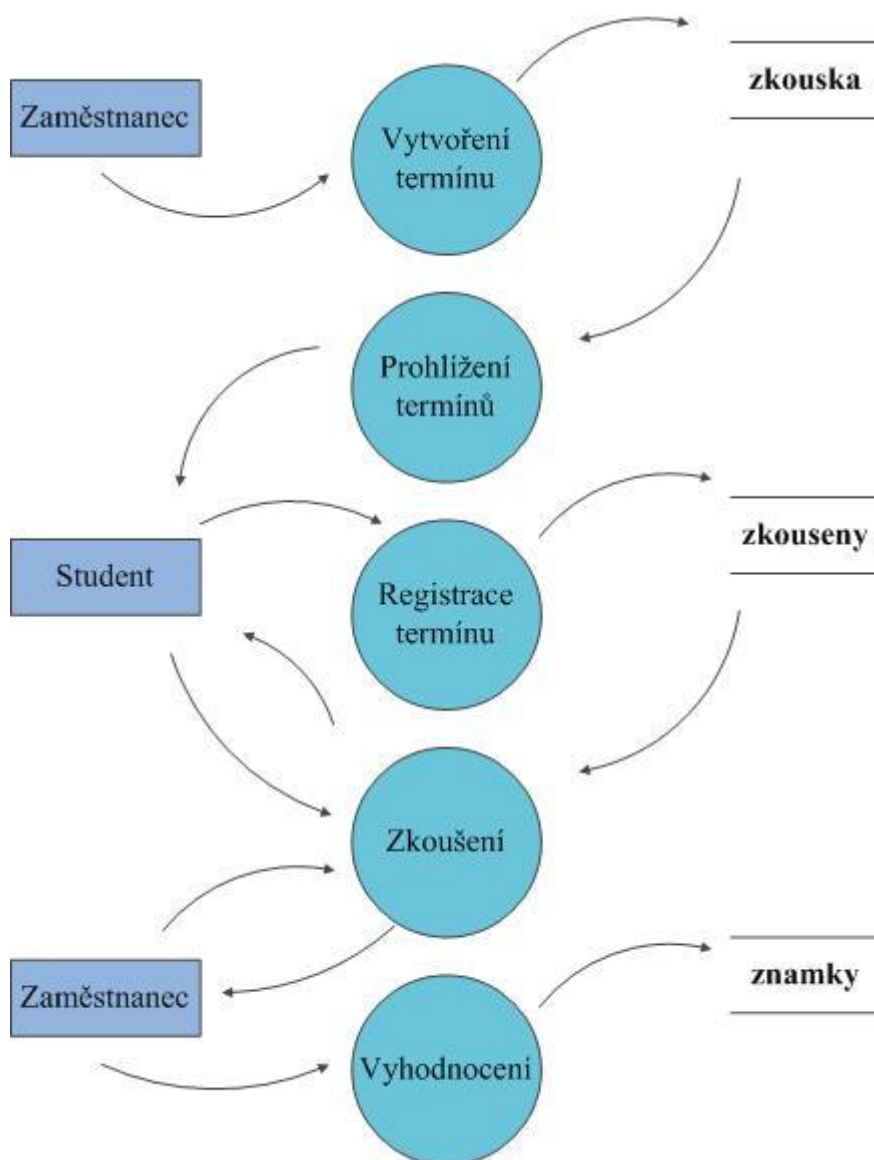


Obrázek 5 – ER diagram
zdroj – vlastní zpracování

4.3.2 Pololetní období

Zkoušky jsou hlavním předmětem, kvůli kterému vzniká celá tato databáze. Na konci každého pololetí budou učitelé vypisovat termíny zkoušek z různých předmětů. Studenti si budou moct registrovat zvolený termín, případně jej následně měnit. Po uskutečnění zkoušky provede zaměstnanec bodové hodnocení, které se zapíše mezi ostatní známky studenta.

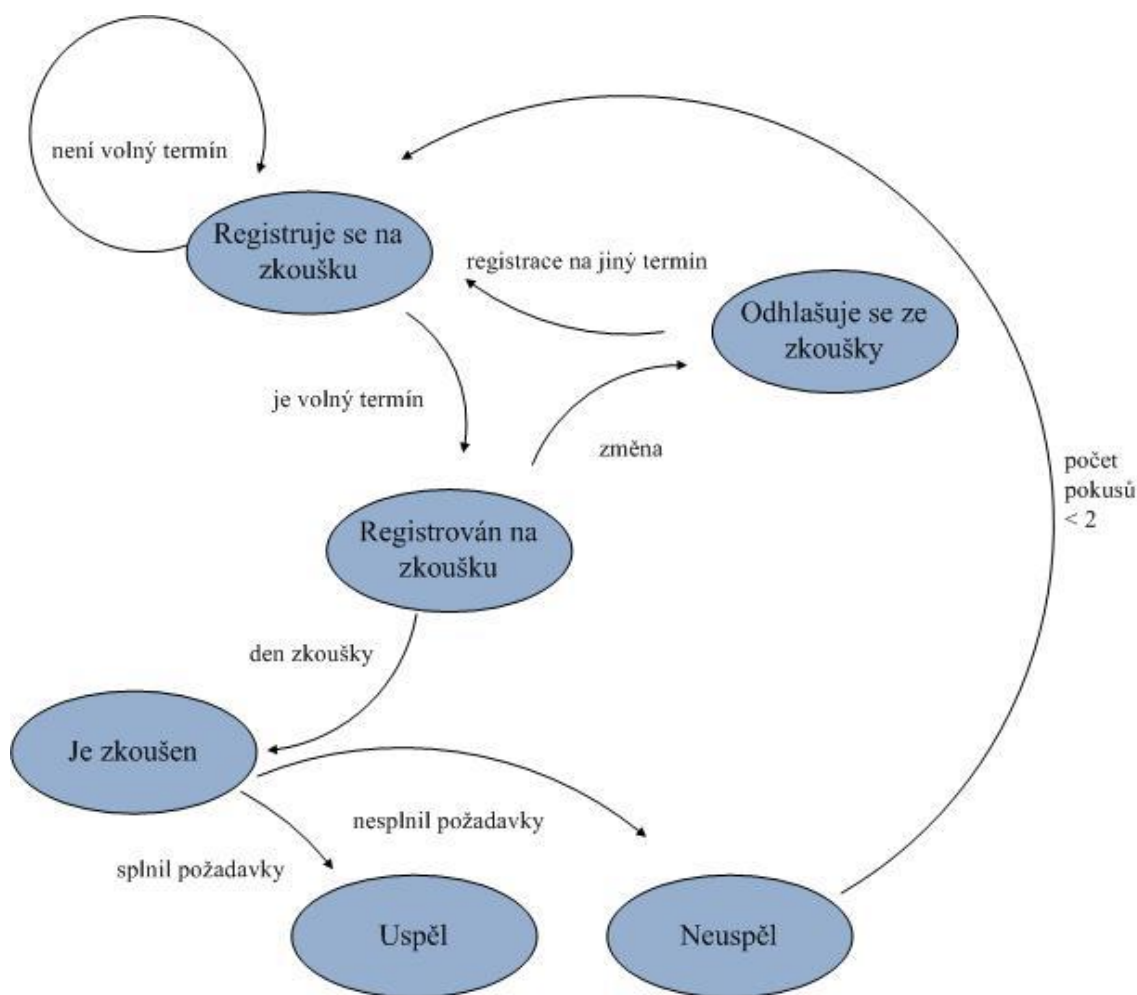
DFD diagram pololetního období



Obrázek 6 – DFD diagram pololetního období
zdroj: vlastní zpracování

Během pololetního období nabývají studenti gymnázia určitých stavů vzhledem ke zkouškám, které je potřeba sledovat z důvodu kontrolování pravidel pro registraci a vykonání zkoušky.

Stavový digram studenta během pololetních zkoušek

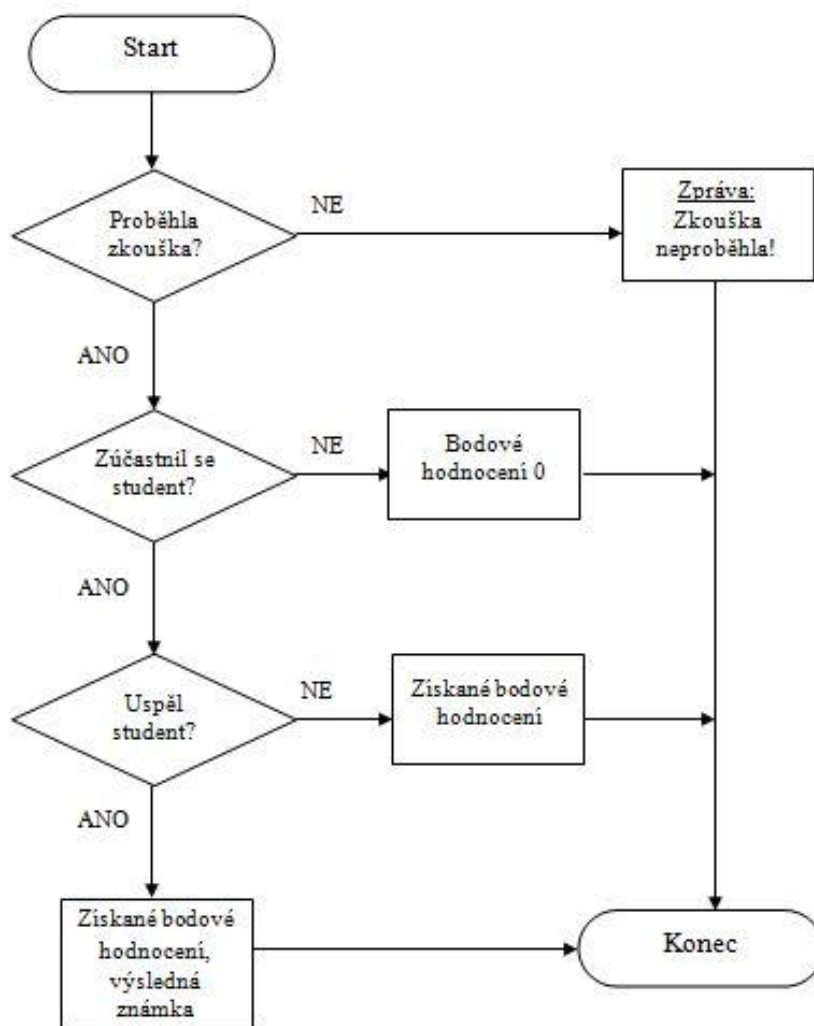


Obrázek 7 – Stavový diagram studenta během pololetních zkoušek
zdroj: vlastní zpracování

Proces vyhodnocení zkoušky

Proces bude mít čtyři možná zakončení. Při neuskutečněné zkoušce se hned ukončí a oznámí zprávou: „Zkouška neproběhla“. Pokud se student nezúčastní zkoušky, proces se ukončí a automaticky bude student ohodnocen nula body. Pokud se student zúčastní

a neuspěje, bude mu uděleno hodnocení v podobě získaných bodů. Při úspěchu u pololetní zkoušky mu budou zapsány body a přidělena výsledná známka, která bude napsána na vysvědčení.



Obrázek 8 – Proces vyhodnocení zkoušky
zdroj: vlastní zpracování

4.4 Logický a fyzický návrh

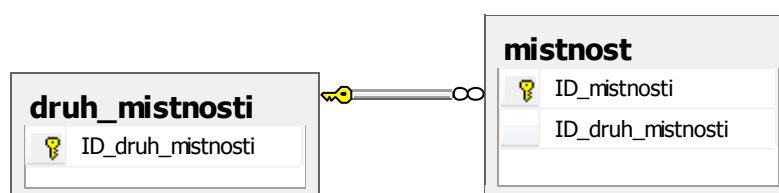
Tato část návrhu bude vycházet z ER diagramu, který byl vytvořen v konceptuální části. Vytvoříme požadované tabulky, zkontrolujeme strukturu, ověříme požadované transakce a integritní omezení.

4.4.1 Tabulky atributů a číselníky

Jako první krok při vytváření databáze je tvorba tabulek, které budou atributy u více částí databáze nebo budou nepřímými atributy – jejich spojení bude tvořit atribut pro určitou tabulku.

tabulky:

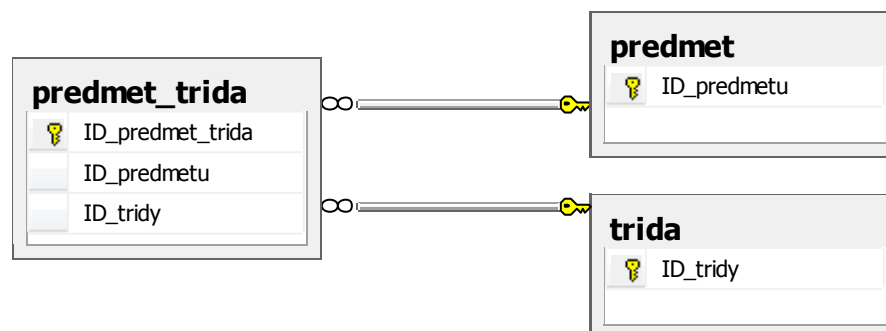
- *PSC* – tvoří číselník všech PSČ a jmen obcí (Rajhrad 66461,...), primárním klíčem je *ID_PSC*
- *oprávnění* – tvoří číselník všech úrovní oprávnění uživatele (administrátor, učitel, student,...), primárním klíčem je *ID_opravneni*
- *druh_mistnosti* – tvoří seznam všech možných druhů místností vyskytujících se v budově školy (učebna, kabinet, sklad,...), primárním klíčem je *ID_druh_mistnosti*
- *mistnost* – zaznamenává informace o místnostech v budově školy (druh, označení, patro a popis), primárním klíčem je *ID_mistnosti*, druh je určován pomocí cizího klíče *ID_druh_mistnosti* z tabulky *druh_mistnosti*



Obrázek 9 – Druh místnosti
zdroj: vlastní zpracování

- *predmet* – tvoří seznam všech vyučovaných předmětů (matematika, český jazyk, fyzika,...), primárním klíčem je *ID_predmetu*
- *trida* – tvoří seznam všech možných tříd gymnázia (prima, sekunda, tercie, 4. A, 4. B,...), primárním klíčem je *ID_tridy*

- *predmet_trida* – tabulka tvořící seznam všech úrovní předmětu (M1, M2, M3, AJ1, AJ8,...), primárním klíčem je *ID_predmet_trida*, úroveň je určována pomocí cizích klíčů z tabulek *predmet* a *trida*



Obrázek 10 – Úroveň předmětu
zdroj: vlastní zpracování

4.4.2 Evidence zaměstnanců

Z hlavních částí databáze jsem jako první zvolil evidenci zaměstnanců. Před vytvořením samotné tabulky zaměstnanců bylo zapotřebí, kromě již vytvořených tabulek, několika dalších, které budou zaměstnance blíže specifikovat.

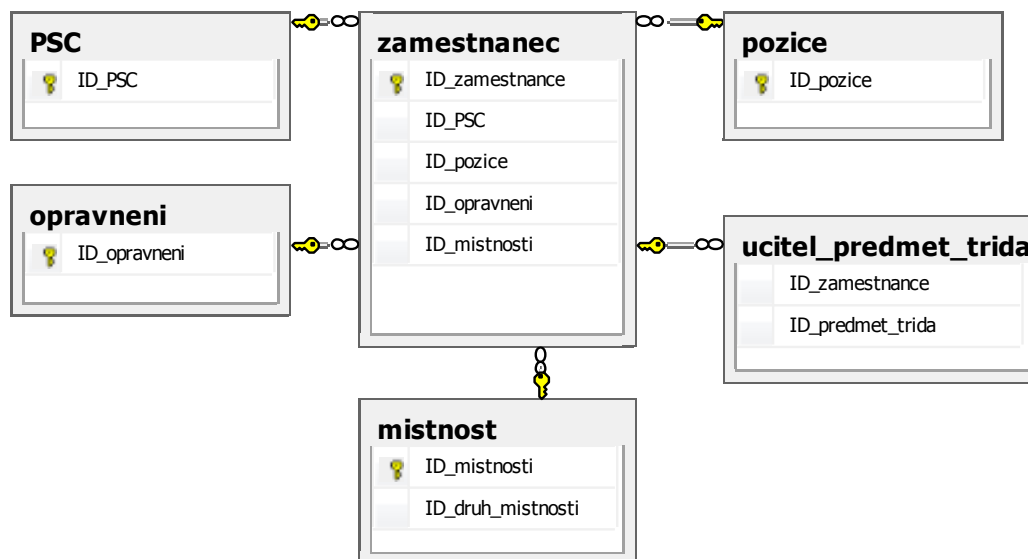
související tabulky:

- *pozice* – tvoří seznam všech možných pozic zaměstnanců školy (ředitel, učitel, sekretářka,...), primárním klíčem je *ID_pozice*
- *ucitel_predmet_trida* – tabulka která přiřazuje učiteli předmět, který vyučuje (matematika pro primu, matematika pro čtvrtý ročník,...), primárním klíčem je dvojice cizích klíčů *ID_predmet_trida* a *ID_zamestnanec*

Zaměstnanec

Tabulka *zamestnanec* je určena primárním klíčem *ID_zamestnanec*, který bude automaticky generován při vzniku záznamu. Dále bude obsahovat osobní údaje jako tituly, jméno, rodné číslo, adresa, číslo účtu, které zaměstnavatel potřebuje pro převod výplaty a další úkony. Každý zaměstnanec bude mít přidělen pracovní mail a telefon. Pokud nebude mít vlastní pracovní telefon, tak zde bude uveden telefon do určené

místnosti. Možností je i soukromý mail a telefon, které nejsou povinným údajem. Pozice zaměstnance je určena cizím klíčem *ID_pozice*. Poslední cizí klíč, *ID_mistnosti* určí místnost, v které by měl být zaměstnanec nalezen během pracovní doby. V tabulce bude taktéž zaznamenáno přihlašovací jméno a heslo uživatele IS.

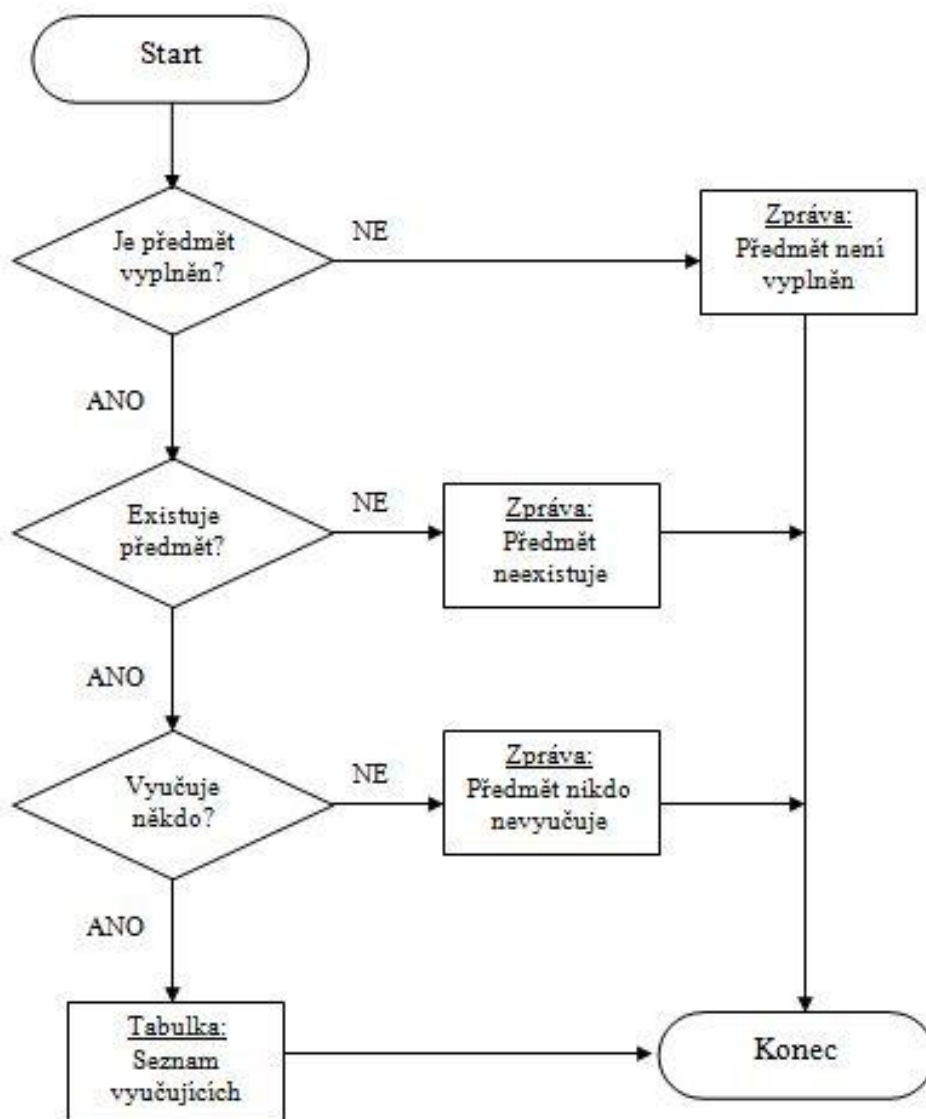


Obrázek 11 – Zaměstnanec
zdroj: vlastní zpracování

Procedura *vyucujici*

K sestavě zaměstnanců jsem navrhnul proceduru *vyucujici*. Navrhovaná procedura bude mít tři vstupní parametry, podle kterých bude vyhledávat. Prvním je název předmětu, který slouží k vyhledání všech zaměstnanců vyučujících zvolený předmět. Při nevyplnění prvního parametru bude procedura vyhledávat podle druhého a třetího (jméno a příjmení zaměstnance). Výsledkem bude výpis všech předmětů, které může zaměstnanec vyučovat. Pokud by nevyhovoval způsob tří vstupních parametrů je možnost proceduru rozdělit na dvě samostatné. Procedura by byla využita například při hledání náhradního vyučujícího pro zvolený předmět.

Vývojový diagram procedury se vstupní proměnou *predmet*



Obrázek 12 – Vývojový diagram procedury
zdroj: vlastní zpracování

Procedura bude mít čtyři možná zakončení. Při nevyplněném vstupním parametru se hned ukončí a napíše zprávu: „*Předmět není vyplněn*“. Pokud vyplněný předmět nebude existovat, procedura se ukončí zprávou: „*Předmět neexistuje*“. Při existenci záznamu předmětu proběhne hledání, jestli je vyučován. V případě, že nebude nalezen

záznam o vyučujícím, procedura se ukončí zprávou: „*Předmět nikdo nevyučuje*“. Posledním možným zakončením je tabulka, která bude obsahovat jména učitelů, kteří vyučují daný předmět.

4.4.3 Evidence studentů a jejich zástupců

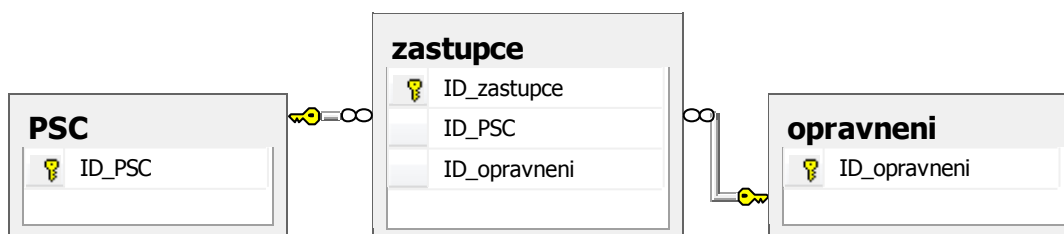
Druhou částí databáze je evidence studentů a jejich zákonných zástupců. Po ukončení studia se informace budou archivovat a přesouvat, z důvodu rychlého vyhledávání mezi aktivními studenty, do jiných tabulek. Kromě již vytvořených tabulek atributů (*PSC*, *opraveni*, *trida*, *predmet*, *predmet_trida*), bude za potřeby několika dalších.

související tabulky:

- *stav* – tvoří číselník všech možných stavů studia (aktivní, přerušeno,...), primárním klíčem je *ID_stavu*
- *rocnik* – tabulka zaznamenávající počátek studia, předpokládané ukončení a délku studia (1990 – 1998 – 8 let, 1990 – 1994 – 4 roky,...), primárním klíčem je *ID_rocniku*
- *student_zastupce* – tabulka spojující studenty s jejich zástupci, primárním klíčem je dvojice cizích klíčů *ID_zastupce* a *ID_student*

Zástupce

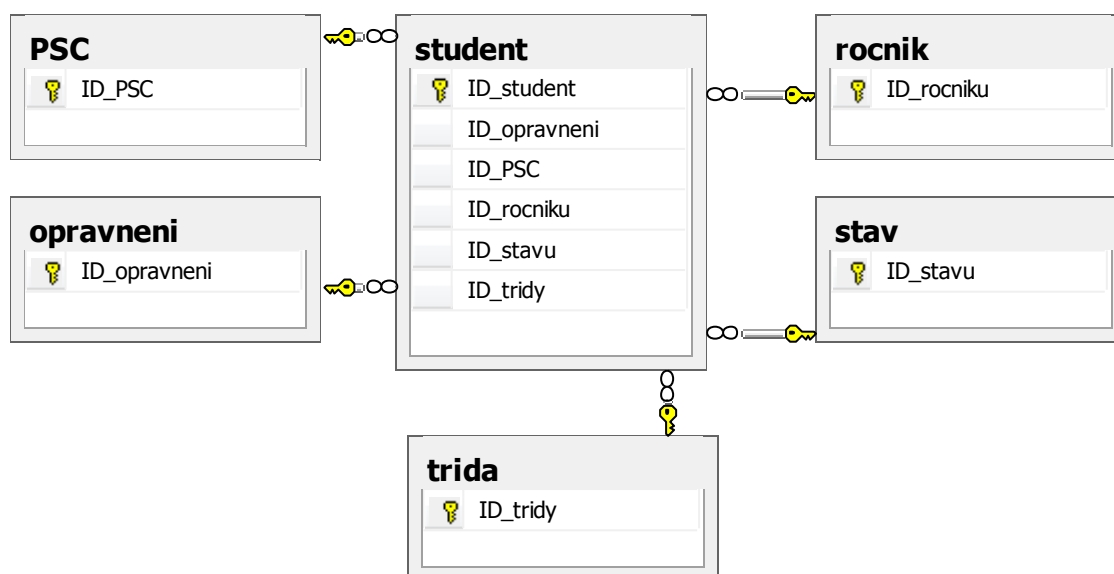
Tabulka *zastupce* je určena primárním klíčem *ID_zastupce*, který bude automaticky generován při vzniku záznamu. Dále bude obsahovat osobní údaje jako tituly, jméno, adresa, telefon, mail, informace o zaměstnání a číslo účtu, které škola potřebuje kvůli kontrole plateb školného. V tabulce bude taktéž zaznamenáno přihlašovací jméno a heslo uživatele IS.



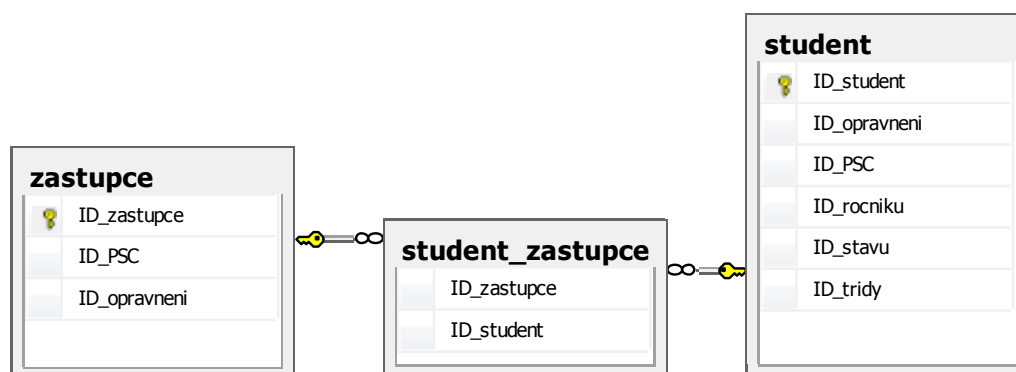
Obrázek 13 – Zástupce
zdroj: vlastní zpracování

Student

Tabulka *student* je určena primárním klíčem *ID_student*, který bude automaticky generován při vzniku záznamu. Dále bude obsahovat osobní údaje jako jméno, rodné číslo, adresu a mail přidělený školou. Ročník studia bude přiřazen cizím klíčem *ID_rocniku* a učební třída pomocí *ID_trída*, přes kterou se budou dále určovat předměty pro daného žáka. V tabulce bude taktéž zaznamenáno přihlašovací jméno a heslo uživatele IS.



Obrázek 14 – Student
zdroj: vlastní zpracování



Obrázek 15 – Spojení zástupce a studenta
zdroj: vlastní zpracování

Procedura *tridniseznam*

Pro školu je důležité mít přehled o svých studentech, k čemuž slouží třídní seznamy. Tato procedura umožní výpis všech studentů vybrané třídy. Záznam bude kvůli přehlednosti seřazen podle příjmení jednotlivých žáků. Do selectu jsem zvolil i zobrazení ročníku do kterého vybraní žáci patří.

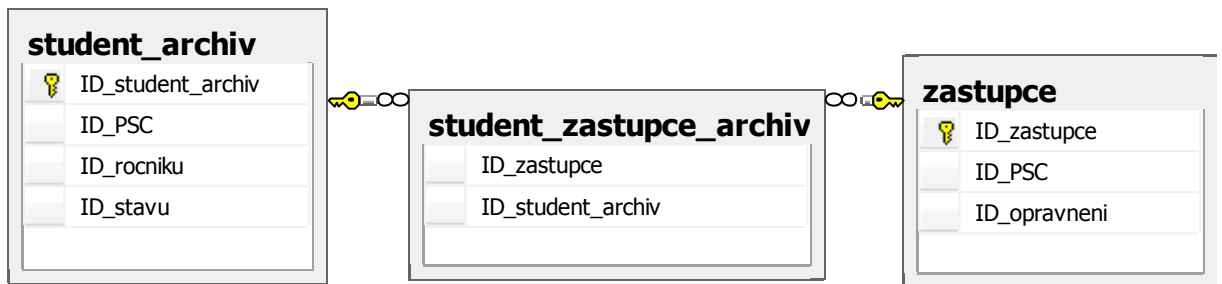
```
create procedure tridniseznam
@trida varchar (8)
as
begin
select s.prijmeni, s.jmeno, t.nazev, r.zahajeni, r.zakonceni, r.delka
from dbo.student s, dbo.trida t, dbo.rocnik r
where t.nazev=@trida and t.ID_tridy=s.ID_tridy and r.ID_rocniku=s.ID_rocniku
order by s.prijmeni
end
go
```

Dotaz na spuštění procedury:

```
execute tridniseznam 'prima'
```

Archivace studentů

K účelu archivace studentů je navržena tabulka *student_archiv*, která bude téměř totožná jako tabulka *student*, pouze bez nepotřebných údajů. Bude se jednat o *ID_tridy*, *ID_opravneni*, přihlašovací jméno a heslo uživatele IS, z kterého budou smazáni. Primárním klíčem bude *ID_studenta_archiv*, které bude stejné, jako bývalo *ID_studenta*. Přiřazení studentů k zástupcům bude probíhat totožně jako při běžné evidenci, pomocí třetí tabulky *student_zastupce_archiv*, primární klíč bude tvořen cizími klíči *ID_studenta_archiv* a *ID_zastupce*.



Obrázek 16 – Spojení archivovaných studentů
zdroj: vlastní zpracování

Procedura *archiv*

V případě zakončení, nebo přerušení studia je potřeba přesunout studenta i s jeho zástupci do tabulek určených pro archivaci. Procedura bude mít jeden vstupní parametr, *ID_studenta*. Proceduru je možné navrhnout pro současnou archivaci celé třídy, která zakončí studium. Dalším možným řešením je trigger, který se automaticky spustí při změně *stavu* v tabulce studenta.

```
create procedure archiv
```

```
@id_studenta int
```

```
as
```

```
begin
```

```
insert into
```

```
dbo.student_archiv(ID_student_archiv,jmeno,prijmeni,rc,mail,ulice,cp,ID_PSC,ID_rocniku, ID_stavu)
```

```
select s.ID_student, s.jmeno, s.prijmeni,s.rc,s.mail,s.ulice,s.cp,s.ID_PSC,s.ID_rocniku, s.ID_stavu
```

```
from dbo.student s
```

```
where s.ID_student=@id_studenta
```

```
insert into dbo.student_zastupce_archiv (ID_student_archiv,ID_zastupce)
```

```
select sa.ID_student_archiv, z.ID_zastupce
```

```
from dbo.zastupce z, dbo.student_archiv sa, dbo.student_zastupce sz, dbo.student s
```

```
where @id_studenta=sa.ID_student_archiv and z.ID_zastupce = sz.ID_zastupce and sz.ID_student = s.ID_student
```

```

delete from dbo.student_zastupce where
dbo.student_zastupce.ID_student=@id_studenta
delete from dbo.student where dbo.student.ID_student=@id_studenta
end
go

```

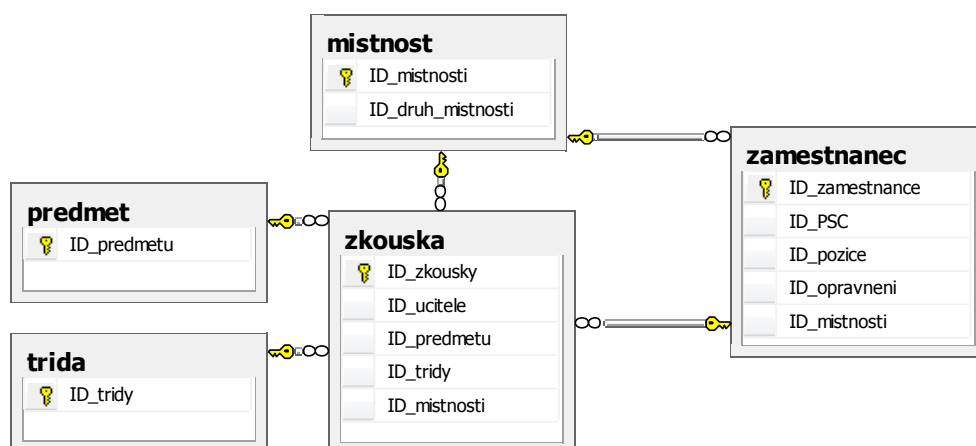
Dotaz na spuštění procedury:

```
execute archiv 1
```

Procedura prvně vytvoří záznam studenta v tabulce *student_archiv*. V tabulce *student_zastupce_archiv* přiřadí ke studentovi jeho zástupce. Následné mazání musí proběhnout kvůli vazbám mezi tabulkami v opačném pořadí. Prvně se smažou veškeré vazby zástupce ke studentovi a až potom samotný student.

4.4.4 Zkouška

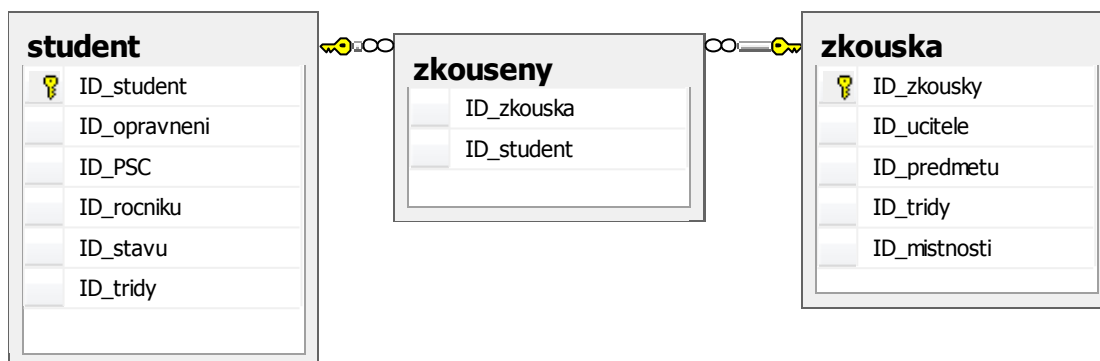
Tabulka *zkouska* je určena primárním klíčem *ID_zkousky*, který bude automaticky generován při vzniku záznamu. Dále bude obsahovat datum a čas konání zkoušky, délku zkoušky, maximální počet možných bodů a maximální kapacita studentů, kteří se zúčastní zkoušky. Místo bude určenou cizím klíčem *ID_mistnosti*. Zaměstnanec, který zkoušku povede je přiřazen cizím klíčem *ID_zamestnance*. Třída a předmět zkoušky bude učen cizím klíčem *ID_trida* a *ID_predmet*.



Obrázek 17 – Zkouška
zdroj: vlastní zpracování

Registrace na zkoušku

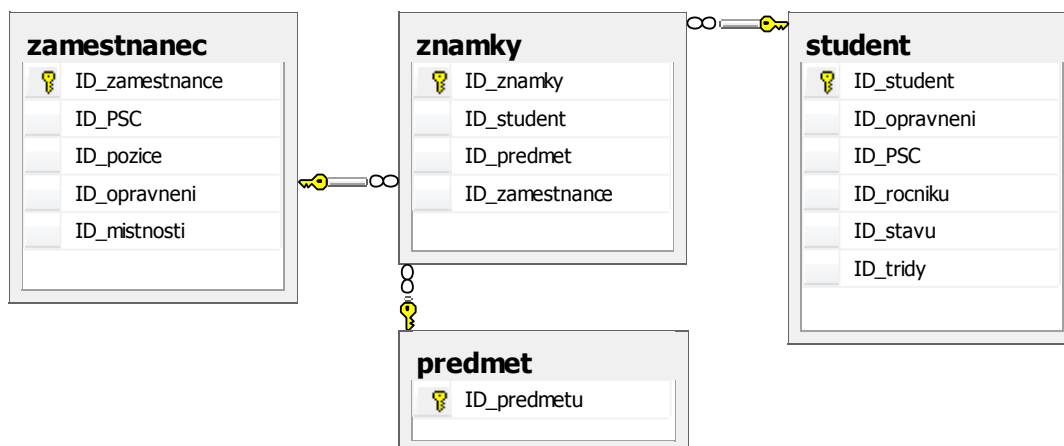
Když si student vybere zkoušku a zaregistruje se na ni, vznikne záznam v tabulce *zkouseny*, která obsahuje dva cizí klíče *ID_studenta* a *ID_zkousky*, které dohromady tvoří primární klíč tabulky.



Obrázek 18 – Zkoušený
zdroj: vlastní zpracování

4.4.5 Hodnocení

Po vykonání zkoušky provede zaměstnanec hodnocení. Výsledek zkoušky bude vložen do tabulky *znamky*, v které budou uloženy veškerá bodová hodnocení získaná během školního roku. Tabulka *znamky* je určena primárním klíčem *ID_znamky*, který bude automaticky generován při vzniku záznamu. Dále bude obsahovat datum a čas vložení hodnocení a maximální počet možných. Předmět, z kterého je známka bude určen cizím klíčem *ID_predmet*. Zaměstnanec, který udělil bodové hodnocení je přiřazen cizím klíčem *ID_zamestnance*. Studenta, kterému hodnocení bude uděleno, přiřazuje cizí klíč *ID_studenta*. Hodnocení probíhá celý rok pouze bodově, až na konci pololetí jsou studentům určeny známky, podle stupnice po deseti procentech.



Obrázek 19 – Znamky
zdroj: vlastní zpracování

Procedura *znamka*

Hlavní zájem rodiče projevují o výsledky svých dětí, které jim přiblíží procedura *znamka*. Bude mít dva vstupní parametry, *ID_studenta* a *ID_predmetu*, které určí studenta a předmět, který nás zajímá. Výsledkem bude zpráva se jménem studenta a předmětu, se získaným a maximálním počtem bodů a výsledné procento, tudíž známka.

```

create procedure znamka
@id_studenta int,
@id_predmetu int
as
begin
declare
@maximum varchar (5),
@body varchar (5),
@procent varchar(4),
@jmeno varchar (50),
@predmet varchar (15)
set @predmet = (select p.nazev from dbo.predmet p where
p.ID_predmetu=@id_predmetu)
  
```

```

set @jmeno = ((select s.jmeno from dbo.student s where @id_studenta=s.ID_student)+'
'+(select s.prijmeni from dbo.student s where @id_studenta=s.ID_student))
set @body = (convert(varchar,(select SUM(z.body) from dbo.znamky z where
@id_studenta=z.ID_student and @id_predmetu=z.ID_predmet)))
set @maximum = (convert(varchar,(select SUM(z.maximum) from dbo.znamky z
where @id_studenta=z.ID_student and @id_predmetu=z.ID_predmet)))
set @procent = (convert(varchar,(select
(convert(float,SUM(z.body)))/(convert(float,SUM(z.maximum)))*100 from
dbo.znamky z where @id_studenta=z.ID_student and @id_predmetu=z.ID_predmet)))
print (@jmeno + ' získal z předmětu ' + @predmet + ' ' + @body + ' bodů z možných ' +
@maximum + ' bodů. ' + 'Výsledné hodnocení je ' + @procent + '%.')
end

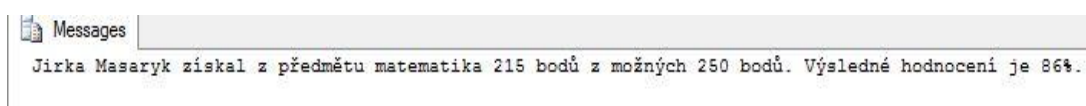
go

```

Dotaz na spuštění procedury:

```
execute znamka 2,1
```

Výsledek procedury :



Obrázek 20 – Celkové hodnocení
zdroj: vlastní zpracování

4.5 Přínos návrhu

Cílem bakalářské práce bylo vytvořit návrh databáze, proto nelze změřit ani vyjádřit číslý přínos. Přínos návrhu bude pouze předpokládaný, protože databáze nebyla v současné době implementována.

předpokládané přínosy:

- zefektivnění procesu registrace na zkoušku
- zkvalitnění komunikace s rodiči a žáky
- zkvalitnění komunikace se zaměstnanci
- zlepšení evidence studentů, zástupců a zaměstnanců
- snadná tvorba seznamů, přehledů
- menší náročnost na zaměstnance
- připravenost pro rozšiřování databáze
- rychlejší prohlížení záznamů o prospěchu studenta

5 Závěr

Cílem mé bakalářské práce bylo navrhnout databázi pro Moravské gymnázium s.r.o., na které by byl budován systém online registrace studentů k pololetním zkouškám. Součástí navrhnuté databáze je i evidence zaměstnanců, studentů gymnázia a jejich zákonných zástupců. V databázi jsou zaznamenávány i známky studentů získané během studia a umožňuje archivaci bývalých studentů. Dalším požadavkem bylo tvorba databáze, kterou bude možné rozšiřovat podle potřeb gymnázia.

Prvním krokem při zpracování cílů byla analýza současné situace a seznámení s fungováním gymnázia a jejich požadavky na databázi. Během tvorby řešení byly všechny milníky návrhu konzultovány a podle potřeb upravovány.

Tato práce splnila stanovené cíle a návrh databáze by měl být schopný usnadnit registraci studentů na zkoušku a zefektivnit uchovávání dat potřebných pro fungování gymnázia. Pro zavedení databáze do provozu je nutné navrhnout potřebnou aplikaci, uživatelské rozhraní a internetové stránky pro registraci na zkoušky.

Seznam použitých zdrojů

Monografické zdroje

(1) CONOLLY, T., BEGG, C., HOLOWCZAK, R. *Mistrovství – databáze:*

Profesionální průvodce tvorbou efektivních databází. 1. vydání. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

(2) GROFF, J., WEINBERG, P. *SQL Kompletní průvodce.* 1. vydání. Brno: Computer Press, 2005. 936 s. ISBN 80-251-0369-2.

(3) HOTEK, M. *Microsoft SQL Server 2008: krok za krokem.* 1. vydání. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6.

(4) HOULETTE, F. *SQL: Příručka programátora.* 1. vydání. Praha: SoftPress, 2001. 382 s. ISBN 80-86497-14-3.

(5) JAMES, M. *Návrh databází.* Praha: Grada, 2006. 408 s. ISBN 80-247-0900-7.

(6) KOCH, M., NEUWIRTH, B. *Datové a funkční modelování.* 3. přepracované vydání. Brno: CERM, 2008. 121 s. ISBN 978-80-214-3731-9.

(7) RIESSLER, P. *Databáze a programování.* Brno: Vysoké učení technické, 2000. 102 s. ISBN 80-214-1778-1.

Internetové zdroje

(8) *Bakaláři*. [online]. 2011 [2012-02-10]. Dostupné na: <<http://www.bakalari.cz/>>.

(9) *MP-Soft a.* [online]. 2007 [2012-02-10]. Dostupné na: <<http://www.mp-soft.cz/>>.

(10) *Skolaonline*. [online]. 2012 [2012-02-10]. Dostupné na: <<http://www.skolaonline.cz/>>.

Seznam obrázků a tabulek

Seznam obrázků

Obrázek 1 – Data, informace a znalosti	12
Obrázek 2 - Vazby mezi entitami	15
Obrázek 3 – Životní cyklus databáze.....	18
Obrázek 4 – Organizační hierarchie	25
Obrázek 5 – ER diagram.....	35
Obrázek 6 – DFD diagram pololetního období	36
Obrázek 7 – Stavový diagram studenta během pololetních zkoušek.....	37
Obrázek 8 – Proces vyhodnocení zkoušky	38
Obrázek 9 – Druh místnosti	39
Obrázek 10 – Úroveň předmětu.....	40
Obrázek 11 – Zaměstnanec.....	41
Obrázek 12 – Vývojový diagram procedury.....	42
Obrázek 13 – Zástupce	43
Obrázek 14 – Student.....	44
Obrázek 15 – Spojení zástupce a studenta.....	44
Obrázek 16 – Spojení archivovaných studentů.....	46
Obrázek 17 – Zkouška	47
Obrázek 18 – Zkoušený	48
Obrázek 19 – Znamky.....	49
Obrázek 20 – Celkové hodnocení	50

Seznam tabulek

Tabulka 1 – Přehled hardwaru	28
Tabulka 2 – Přehled IS.....	31
Tabulka 3 – Identifikace entit	34

Seznam příloh

Příloha 1 – Datový slovník	58
Příloha 2 – ER diagram.....	62
Příloha 3 – Zdrojový kód	63

Příloha 1 – Datový slovník

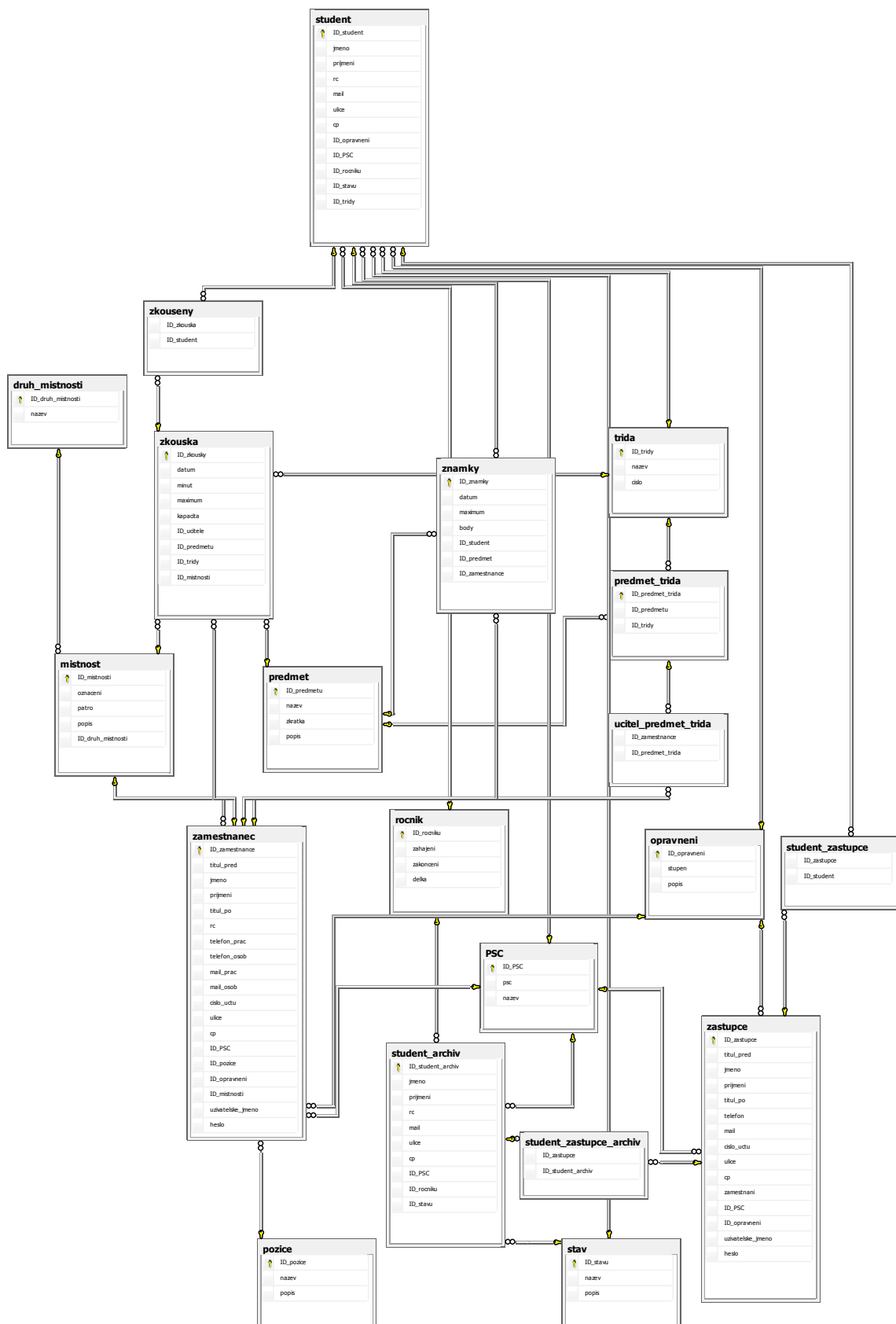
Tabulka	Atributy	Datový typ	Délka	Klíč	Not NULL
druh_mistnosti	ID_druh_mistnosti	int		PK	x
	nazev	varchar	15		x
mistnost	ID_mistnosti	int		PK	x
	oznaceni	varchar	4		x
	patro	numeric	1;0		x
	popis	text			
	ID_druh_mistnosti	int		FK	x
opraveni	ID_opraveni	int		PK	x
	stupeň	int	1		x
	popis	text			
pozice	ID_pozice	int		PK	x
	nazev	varchar	30		x
	popis	text			
predmet	ID_predmetu	int		PK	x
	nazev	varchar	40		x
	zkratka	varchar	5		x
	popis	text			
predmet_trida	ID_predmet_trida	int		PK	x
	ID_predmetu	int		FK	x
	ID_tridy	int		FK	x
PSC	ID_PSC	int		PK	x
	psc	numeric	5;0		x
	nazev	varchar	40		x
rocnik	ID_rocniku	int		PK	x
	zahajeni	date			x
	zakonceni	date			x
	delka	numeric	1;0		x
stav	ID_stavu	int		PK	x
	nazev	varchar	10		x
	popis	text			

Tabulka	Atributy	Datový typ	Délka	Klíč	Not NULL
student	ID_student	int		PK	x
	jmeno	varchar	20		x
	prijmeni	varchar	30		x
	rc	numeric	10		x
	mail	varchar	50		x
	ulice	varchar	50		x
	cp	int			x
	ID_opravneni	int		FK	x
	ID_PSC	int		FK	x
	ID_rocniku	int		FK	x
	ID_stavu	int		FK	x
	ID_tridy	int		FK	x
student_archiv	ID_student_archiv	int		PK	x
	jmeno	varchar	20		x
	prijmeni	varchar	30		x
	rc	numeric	10		x
	mail	varchar	50		x
	ulice	varchar	50		x
	cp	int			x
	ID_PSC	int		FK	x
	ID_rocniku	int		FK	x
	ID_stavu	int		FK	x
student_zastupce	ID_zastupce	int		FK, PK	x
	ID_student	int		FK, PK	x
student_zastupce_archiv	ID_zastupce	int		FK, PK	x
	ID_student_archiv	int		FK, PK	x
trida	ID_tridy	int		PK	x
	nazev	varchar			x
	cislo	numeric	1;0		x
ucitel_predmet_trida	ID_zamestnance	int		FK, PK	x
	ID_predmet_trida	int		FK, PK	x

Tabulka	Atributy	Datový typ	Délka	Klíč	Not NULL
zamestnanec	ID_zamestnanec	int		PK	x
	titul_pred	varchar	10		
	jmeno	varchar	20		x
	prijmeni	varchar	30		x
	titul_po	varchar	10		
	rc	numeric	10;0		x
	telefon_prac	numeric	9;0		x
	telefon_osob	numeric	9;0		x
	mail_prac	varchar	50		x
	mail_osob	varchar	50		
	cislo_uctu	varchar	15		x
	ulice	varchar	50		x
	cp	int			x
	ID_PSC	int		FK	x
	ID_pozice	int		FK	x
	ID_opravneni	int		FK	x
	ID_mistnosti	int		FK	x
	uzivatelske_jmeno	varchar	20		x
	heslo	varchar	20		x
zastupce	ID_zastupce	int		PK	x
	titul_pred	varchar	10		
	jmeno	varchar	20		x
	prijmeni	varchar	30		x
	titul_po	varchar	10		
	telefon	numeric	9;0		x
	mail	varchar	50		
	cislo_uctu	varchar	15		
	ulice	varchar	50		x
	cp	int			x
	zamestnani	varchar	50		
	ID_PSC	int		FK	x
	ID_opravneni	int		FK	x
	uzivatelske_jmeno	varchar	20		x
	heslo	varchar	20		x

Tabulka	Atributy	Datový typ	Délka	Klíč	Not NULL
zkouseny	ID_zkouska	int		FK, PK	x
	ID_student	int		FK, PK	x
zkouska	ID_zkouska	int		PK	x
	datum	datetime			x
	minut	int			x
	maximum	int			x
	kapacita	int			x
	ID_ucitele	int		FK	x
	ID_predmetu	int		FK	x
	ID_tridy	int		FK	x
	ID_mistnosti	int		FK	x
znamky	ID_znamky	int		PK	x
	datum	date			x
	maximum	int			x
	body	int			x
	ID_student	int		FK	x
	ID_predmet	int		FK	x
	ID_zamestnance	int		FK	x

Příloha 2 – ER diagram



Příloha 3 – Zdrojový kód

```
create database MG

use MG
go

--číselník PSČ
create table PSC
(
ID_PSC int identity (1,1) primary key not null,
psc numeric (5) not null,
nazev varchar (40) not null
)

--číselník stavu studia
create table stav
(
ID_stavu int identity (1,1) primary key not null,
nazev varchar (10) not null,
popis text
)

--číselník tříd
create table trida
(
ID_tridy int identity (1,1) primary key not null,
nazev varchar (8) not null,
cislo numeric (1) not null
)

--číselník studijních ročníků
create table rocnik
(
ID_rocniku int identity (1,1) primary key not null,
zahajeni date not null,
zakonceni date not null,
delka numeric (1) not null
)
```

--číselník pozic

create table pozice

```
(
ID_pozice int identity (1,1) primary key not null,
nazev varchar (30) not null,
popis text
)
```

--číselník oprávnění uživatelů

create table opraveni

```
(
ID_opraveni int identity (1,1) primary key not null,
stupen numeric (1) not null,
popis text
)
```

--číselník předmětů

create table predmet

```
(
ID_predmetu int identity (1,1) primary key not null,
nazev varchar (40) not null,
zkratka varchar (5) not null,
popis text
)
```

--číselník druhů místností

create table druh_mistnosti

```
(
ID_druh_mistnosti int identity (1,1) primary key not null,
nazev varchar (15) not null
)
```

--tabulka místností v gymnáziu

create table mistnost

```
(
ID_mistnosti int identity (1,1) primary key not null,
oznaceni varchar (4) not null,
patro numeric (1) not null,
popis text,
```

```
ID_druh_mistnosti int foreign key references druh_mistnosti(ID_druh_mistnosti) not null
)
```

--tabulka zaměstnanců školy

```
create table zamestnanec
(
ID_zamestnanec int identity (1,1) primary key not null,
titul_pred varchar (10),
jmeno varchar (20) not null,
prijmeni varchar (30) not null,
titul_po varchar (10),
rc numeric (10) not null,
telefon_prac numeric (9),
telefon_osob numeric (9) not null,
mail_prac varchar (50),
mail_osob varchar (50),
cislo_uctu varchar (15) not null,
ulice varchar (50) not null,
cp int not null,
ID_PSC int foreign key references PSC(ID_PSC) not null,
ID_pozice int foreign key references pozice(ID_pozice) not null,
ID_opravneni int foreign key references opraveni(ID_opravneni) not null,
ID_mistnosti int foreign key references mistnost(ID_mistnosti) not null,
uzivatelske_jmeno varchar (20) not null,
heslo varchar (20) not null
)
```

--tabulka přiřazující předměty třídě

```
create table predmet_trida
(
ID_predmet_trida int identity (1,1) primary key not null,
ID_predmetu int foreign key references predmet(ID_predmetu) not null,
ID_tridy int foreign key references trida(ID_tridy) not null
)
```

--tabulka pro přiřazení učiteli vybraného předmětu v daném ročníku

```
create table ucitel_predmet_trida
(
ID_zamestnanec int foreign key references zamestnanec(ID_zamestnanec) not null,
```

```
ID_predmet_trida int foreign key references predmet_trida(ID_predmet_trida) not null
)
```

--tabulka zákonných zástupců žáků/studentů

```
create table zastupce
(
ID_zastupce int identity (1,1) primary key not null,
titul_pred varchar (10),
jmeno varchar (20) not null,
prijmeni varchar (30) not null,
titul_po varchar (10),
telefon numeric (9) not null,
mail varchar (50),
cislo_uctu varchar (15),
ulice varchar (50) not null,
cp int not null,
zamestnani varchar (50),
ID_PSC int foreign key references PSC(ID_PSC) not null,
ID_opravneni int foreign key references opravneni(ID_opravneni) not null,
uzivatelske_jmeno varchar (20) not null,
heslo varchar (20) not null
)
```

--tabulka studentů

```
create table student
(
ID_student int identity (1,1) primary key not null,
jmeno varchar (20) not null,
prijmeni varchar (30) not null,
rc numeric (10) not null,
mail varchar (50) not null,
ulice varchar (50) not null,
cp int not null,
ID_opravneni int foreign key references opravneni(ID_opravneni) not null,
ID_PSC int foreign key references PSC(ID_PSC) not null,
ID_rocniku int foreign key references rocnik(ID_rocniku) not null,
ID_stavu int foreign key references stav(ID_stavu) not null,
ID_tridy int foreign key references trida(ID_tridy)
)
```

--tabulka přiřazující studenta k zástupci

```
create table student_zastupce
(
ID_zastupce int foreign key references zastupce(ID_zastupce) not null,
ID_student int foreign key references student(ID_student) not null
)
```

--tabulka archivovaných studentů

```
create table student_archiv
(
ID_student_archiv int primary key not null,
jmeno varchar (20) not null,
prijmeni varchar (30) not null,
rc numeric (10) not null,
mail varchar (50) not null,
ulice varchar (50) not null,
cp int not null,
ID_PSC int foreign key references PSC(ID_PSC) not null,
ID_rocniku int foreign key references rocnik(ID_rocniku) not null,
ID_stavu int foreign key references stav(ID_stavu) not null
)
```

--tabulka přiřazující archivovaného studenta k zástupci

```
create table student_zastupce_archiv
(
ID_zastupce int foreign key references zastupce(ID_zastupce) not null,
ID_student_archiv int foreign key references student_archiv(ID_student_archiv) not null
)
```

--tabulka vytvořených zkoušek

```
create table zkouska
(
ID_zkousky int identity (1,1) primary key not null,
datum datetime not null,
minut int not null,
maximum int not null,
kapacita int not null,
ID_ucitele int foreign key references zamestnanec(ID_zamestnance) not null,
ID_predmetu int foreign key references predmet(ID_predmetu) not null,
)
```

```
ID_tridy int foreign key references trida (ID_tridy) not null,
ID_mistnosti int foreign key references mistnost(ID_mistnosti) not null
)
```

--tabulka přiděluje studenty na zkoušku

```
create table zkouseny
(
ID_zkouska int foreign key references zkouska(ID_zkousky) not null,
ID_student int foreign key references student(ID_student) not null,
)
```

--tabulka známek studentů

```
create table znamky
(
ID_znamky int identity (1,1) primary key not null,
datum date not null,
maximum int not null,
body int not null,
ID_student int foreign key references student(ID_student)not null,
ID_predmet int foreign key references predmet(ID_predmetu) not null,
ID_zamestnance int foreign key references zamestnanec(ID_zamestnance) not null
)
```