



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

# NÁVRH DATABÁZE PRO JAZYKOVOU ŠKOLU

DATABASE DESIGN FOR LANGUAGE SCHOOL

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Lenka Jopeková

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2020

# Zadání bakalářské práce

Ústav:	Ústav informatiky
Studentka:	<b>Lenka Joppeková</b>
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Manažerská informatika
Vedoucí práce:	<b>Ing. Jan Luhan, Ph.D., MSc</b>
Akademický rok:	2019/20

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

## Návrh databáze pro jazykovou školu

### Charakteristika problematiky úkolu:

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

### Cíle, kterých má být dosaženo:

Navrhnout databázi dle požadavků konkrétního subjektu, který působí v oblasti jazykového vzdělávání. Daný návrh bude plně reflektovat na potřeby a základní procesy subjektu se zaměřením na efektivní práci s daty.

### Základní literární prameny:

BEGG, C., R. HOLOWCZAK a T. CONOLLY. Mistrovství - Databáze : Profesionální průvodce tvorbou efektivních databází. Praha: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KROENKE, D. M. a D. J. AUER. Databáze. 6. vyd. Brno: Computer Press, 2015. 496 s. ISBN 978-8-251-4352-0.

LACKO, L. Mistrovství v Microsoft SQL Server 2012. 1 vyd. Praha: Computer Press, 2013. 640 s. ISBN 978-80-251-3773-4.

LAURENČÍK, M. SQL: podrobný průvodce uživatele. Praha: Grada Publishing, 2018. 216 s. ISBN 978-80-271-07742.

LAURENČÍK, M. Tvorba www stránek v HTML a CSS. Praha: Grada Publishing, 2019. 224 s. ISBN 978-80-271-2241-7.

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

V Brně dne 29.2.2020

L. S.

---

doc. RNDr. Bedřich Půža, CSc.  
ředitel

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
děkan

## **Abstrakt**

Táto bakalárska práca obsahuje návrh databázového systému pre jazykovú školu. Cieľom práce je vytvorenie databázového systému, ktorý umožní efektívnejšie a rýchlejšie pracovať s dátami. Obsahom práce sú teoretické východiská pre pochopenie problematiky, analýza súčasného stavu a vlastný návrh databázového systému.

## **Abstract**

This Bachelor's thesis contains a design of a database system for language school. The aim of the thesis is to create a database system, that would enable to process data more effective and faster. The content include theoretical resources of the thesis for understanding of the problematics, analysis of the current condition and the design of a database system.

## **Kľúčové slová**

databáza, databázový systém, SQL, ER – diagram, relácie

## **Key words**

database, database system, SQL, ER – diagram, relations

### **Bibliografická citácia**

JOPPEKOVÁ, Lenka. *Návrh databáze pro jazykovou školu*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/127708>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

## **Čestné prehlásenie**

Prehlasujem, že predložená bakalárska práca je pôvodná a spracovala som ju samostatne. Prehlasujem, že citácie použitých prameňov sú úplné, že som vo svojej práci neporušila autorské práva (v zmysle Zákona č.121/2000 Sb., o práve autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 17.5.2020

.....

podpis

## **Pod'akovanie**

Rada by som pod'akovala vedúcemu práce Ing. Janovi Luhanovi, Ph.D., MSc za odborné vedenie, konzultácie a cenné rady pri spracovávaní tejto bakalárskej práce. Ďalej by som chcela pod'akovať mojim najbližším za veľkú podporu a trpezlivosť v priebehu tvorby práce.

# OBSAH

ÚVOD.....	9
CIELE PRÁCE, METODIKY A POSTUPY SPRACOVANIA.....	10
1 TEORETICKÉ VÝCHODISKÁ PRÁCE .....	11
1.1.    Základné pojmy.....	11
1.1.1.    Informácia.....	11
1.1.2.    Dáta.....	12
1.1.3.    Znalosti .....	13
1.2.    Databázový systém.....	13
1.3.    Dátové modely .....	14
1.3.1.    Lineárny dátový model .....	15
1.3.2.    Hierarchický dátový model.....	16
1.3.3.    Sieťový dátový model.....	16
1.3.4.    Relačný dátový model .....	17
1.3.5.    Objektový dátový model.....	18
1.4.    Relačný dátový model.....	18
1.4.1.    Entity.....	18
1.4.2.    Atribúty.....	19
1.4.3.    Kľúče .....	19
1.4.4.    Relácie .....	19
1.4.5.    Integritné obmedzenie pre vzťahy medzi reláciami.....	20
1.5.    Normálne formy .....	22
1.5.1.    1. normálna forma (multizávislosť).....	22
1.5.2.    2. normálna forma (funkčná závislosť).....	23
1.5.3.    3. normálna forma (tranzitívna závislosť) .....	25
1.6.    Návrh databázy.....	26
1.6.1.    Konceptuálny návrh.....	26
1.6.2.    Logický návrh .....	26
1.6.3.    Fyzický návrh .....	27



1.7.	Jazyk SQL .....	27
1.7.1.	Kategórie príkazov .....	28
1.7.2.	Jazyk DDL .....	28
1.7.3.	Jazyk DQL .....	28
1.7.4.	Jazyk DML .....	29
1.7.5.	Jazyk DCL .....	29
1.7.6.	Príkazy riadenia transakcií .....	29
2	ANALÝZA SÚČASNÉHO STAVU .....	30
2.1.	Základné informácie o jazykovej škole .....	30
2.2.	Organizačná štruktúra .....	30
2.3.	Popis súčasnej situácie .....	31
2.4.	Prihlasovanie študentov .....	32
2.5.	Evidencia študentov .....	34
3	VLASTNÉ NÁVRHY RIEŠENIA .....	35
3.1.	Konceptuálny návrh .....	35
3.1.1.	Identifikácia entít .....	35
3.1.2.	Identifikácia relácií .....	36
3.2.	Logický návrh .....	37
3.2.1.	Tabuľka Zamestnanec .....	37
3.2.2.	Tabuľka Študent .....	38
3.2.3.	Tabuľka Kurz .....	39
3.2.4.	Tabuľka Typ kurzu .....	40
3.2.5.	Tabuľka Jazyk .....	41
3.2.6.	Tabuľka Úroveň .....	41
3.2.7.	Tabuľka Miestnosť .....	42
3.2.8.	Tabuľka Pozícia .....	42
3.2.9.	Tabuľka Deň .....	43
3.2.10.	Tabuľka Veková kategória .....	43
3.2.11.	Tabuľka PSČ .....	44
3.2.12.	Dekompozičná tabuľka Zapísaný kurz .....	44
3.2.13.	Dekompozičná tabuľka Vyučovaný jazyk .....	45
3.2.14.	Dekompozičná tabuľka Pozícia zamestnanca .....	46

3.2.15.	Dekompozičná tabuľka Znalosť jazyka .....	47
3.2.16.	ER Diagram .....	48
3.3.	Fyzický návrh.....	50
3.3.1.	Vytvorenie databázy.....	50
3.3.2.	Vytvorenie tabuliek.....	50
3.3.3.	Definovanie cudzích kľúčov .....	51
3.3.4.	Vloženie testovacích dát .....	51
3.3.5.	Pohľady .....	52
3.3.6.	Trigger.....	55
3.4.	Zhodnotenie návrhu .....	56
ZÁVER	.....	57
ZOZNAM POUŽITÝCH ZDROJOV	.....	58
ZOZNAM POUŽITÝCH SKRATIEK A SYMBOLOV	.....	59
ZOZNAM OBRÁZKOV	.....	60
ZOZNAM TABULIEK	.....	61
ZOZNAM PRÍLOH	.....	62

## ÚVOD

V dnešnej dobe každá firma uchováva čoraz viac údajov potrebných k dobrému fungovaniu firmy, z toho dôvodu je dôležité akým spôsobom dané dáta eviduje a spracováva. Tieto údaje boli uchovávané v papierovej podobe, ale spracovávanie dát a získavanie podstatných informácií začalo byť čoraz zložitejšie s každodenným zväčšovaním množstva zaznamenávaných dát. Z toho dôvodu vznikli elektronické databázy, ktoré sa stali bežnou súčasťou denného života podobne ako počítače.

Databázové systémy vo veľkej miere zefektívnilo a zrýchlilo prácu s obrovským množstvom dát. Väčším či menším firmám a aj niektorým domácnostiam zásadne uľahčujú fungovanie a administráciu na dennej báze. Dobrý systém umožňuje bez ťažkostí evidovať firemné údaje o zamestnancoch, zákazníkoch, službách alebo tovare, ako aj vyhľadávať potrebné informácie.

Každá firma má odlišný prístup spracovania a ukladania nazhromaždených dát. Záleží len na spoločnosti, či zostane verná papierovej podobe alebo prejde na moderné technológie s vybudovaním vlastnej databázy. Pri zostavovaní databázy sa môžu rozhodnúť, ktoré informácie sú pre nich dôležité a či je potrebné ich evidovať. V dnešnej dobe je možné vybrať si z veľkého množstva databázových systémov v rôznych cenových hladinách, kde nájdeme aj finančne dostupné a zároveň užívateľsky prívetivé produkty.

## **CIELE PRÁCE, METODIKY A POSTUPY SPRACOVANIA**

Hlavným cieľom tejto práce je tvorba návrhu databázového systému pre jazykovú školu. Databáza by mala slúžiť k efektívnejšej práci s evidovanými dátami študentov, zamestnancov a kurzov. K dosiahnutiu cieľa bude potrebné zohľadniť požiadavky jazykovej školy.

Na dosiahnutie cieľa bude potrebné dosiahnuť nasledujúce čiastkové ciele:

- spracovanie teoretických východísk problematiky databázových systémov pomocou dostupných knižných zdrojov,
- spracovanie analýzy súčasného stavu spracovávania a ukladania dát v jazykovej škole,
- návrh vlastného riešenia.

Pri spracovávaní tejto bakalárskej práce boli použité rôzne postupy a metódy, vďaka ktorým bolo možné úspešne dokončiť túto záverečnú prácu. Prvým z postupov bolo naštudovanie dostupnej odbornej literatúry pre pochopenie problematiky. Ďalej bolo potrebné konzultovať súčasný stav a požiadavky na databázu s vedením školy. V neposlednej rade bolo potrebné vytvoriť samotný návrh databázy zloženého z konceptuálneho, logického a fyzického návrhu.

# 1 TEORETICKÉ VÝCHODISKÁ PRÁCE

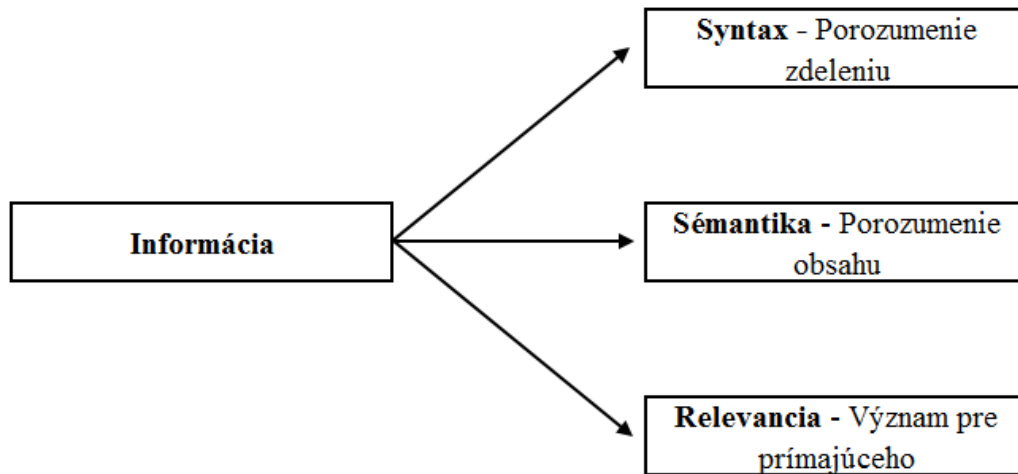
V prvej kapitole sú spracované teoretické východiská tejto práce. Sú tu vysvetlené základné pojmy spojené s databázovými systémami, ktoré sú nutné k porozumeniu danej problematiky. Na začiatku sú základné pojmy pre prácu s databázami, následne aj databázový systém a dátové modely. Záverečná časť teórie obsahuje normálne formy a vyhľadávací jazyk SQL.

## 1.1. Základné pojmy

Medzi základné pojmy dátového modelovania, ktorým by sme mali rozumieť, patria informácie, dáta a znalosti. Tieto termíny sú objasnené v nasledujúcich riadkoch.

### 1.1.1. Informácia

Informáciu môžeme chápať ako správu alebo vnem spĺňajúci tri základné charakteristiky, ktoré zobrazuje obrázok č. 1. Prvou charakteristikou je syntaktická relevancia. To znamená, že subjekt, ktorý informáciu prijíma, musí byť schopný ju zachytiť a rozumieť jej. Nasleduje sémantická relevancia, ktorá znamená, že príjemca musí vedieť čo obsah správy znamená a vypovedá o ňom a jeho okolí. Poslednou požiadavkou je pragmatická relevancia. Pre príjemcu musí mať správa význam (1, s. 4).

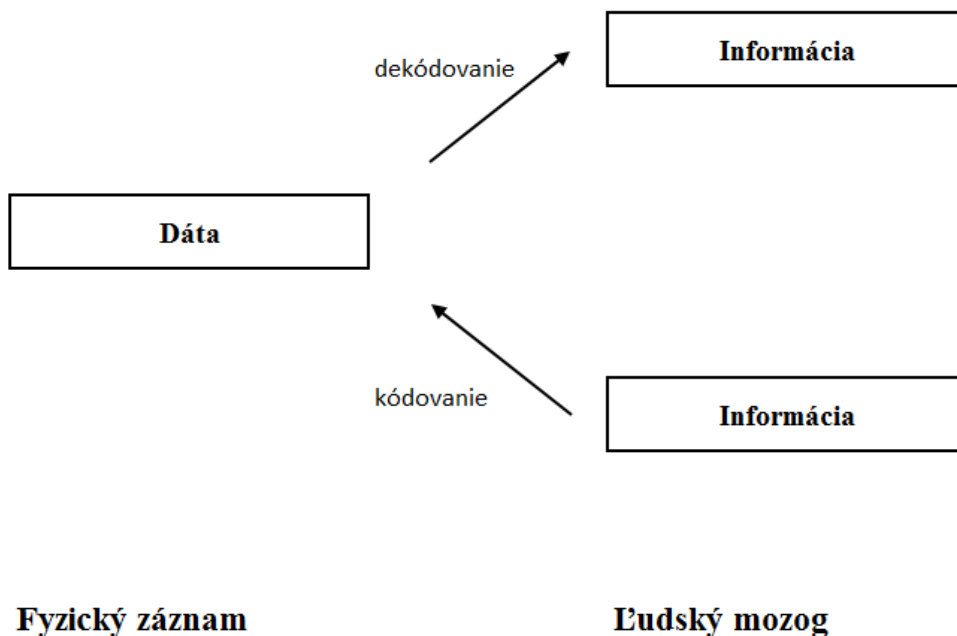


**Obrázok 1: Znázornenie informácie**

(Zdroj: vlastné spracovanie podľa 1, s. 4)

### **1.1.2. Dáta**

Pod pojmom dáta rozumieme fakty, ktoré sa dajú spracovať a sú nositeľmi významu pre ľudí. V prípade využívania dát k rozhodovaniu sa stávajú informáciou, lebo im priradíme význam a zmysel. Z toho dôvodu sa stáva, že dátam sa priradzuje nielen význam správ ale aj informácia. Preto môžeme povedať, že dáta sú potenciálnymi informáciami. Informácie, ktoré ľudia zachytia a porozumejú im nazývame dáta, tie následne môžeme ďalej spracovávať, transformovať do inej podoby a zaznamenať na papier alebo do počítača. Znázornenie dát môžeme vidieť na obrázku č. 2 (1, s. 5).



**Obrázok 2: Znáozornenie dát**

(Zdroj: Vlastné spracovanie podľa 1, s.5)

### 1.1.3. Znalosti

Výsledkom porozumenia informácie a jej integrácie s predchádzajúcimi informáciami je znalosť. Taktiež ju možno chápať ako informácie o tom ako chápať a využívať ďalšie informácie a dáta (aj vo vzájomných kombináciách) (1, s. 5).

## 1.2. Databázový systém

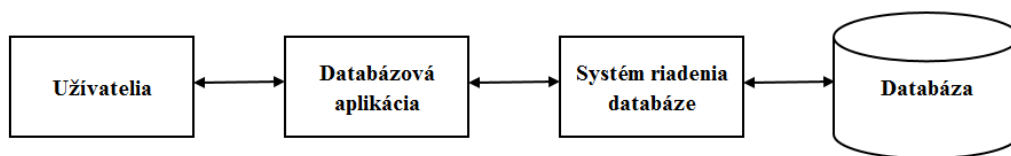
Prostredníctvom databázových systémov sme schopní zhromažďovať všetky pre nás potrebné informácie, či už v škole, práci alebo živote. Požadované informácie ukladá a udržuje centrálné na jednom mieste (2, s. 3).

Tento systém tvoria celkovo štyri časti a tými sú:

- užívateľia,
- databázová aplikácia,

- systém riadenia databázy,
- databáza (3, s. 29).

Vyššie spomenuté komponenty sú zobrazené na obrázku č. 3. Ako prvý komponent z pravej strany je databáza, ktorú môžeme opísať ako súhrn súvisiacich tabuliek a iných štruktúr. Nasleduje systém riadenia databázy, čo je program na vytváranie, spracovávanie a spravovanie databázy. Prijímanie požiadaviek prebieha v jazyku SQL a zadané požiadavky premieňa na aktivity v databáze. Ako treťou zložkou je databázová aplikácia. Je to súbor jedného alebo viacerých počítačových programov, ktoré fungujú ako sprostredkovateľ medzi užívateľom a systémom riadenia databázy. Aplikáčne programy čítajú a upravujú dáta v databázy pomocou SQL príkazov, ktoré posielajú systému riadenia databázy. Takisto prezentujú užívateľom dáta v podobe zostáv a formulárov. Užívatelia si udržiavajú prehľad o určitých skutočnostiach v databáze pomocou databázovej aplikácie. Načítavajú a zadávajú žiadané dáta a vytvárajú dotazy a zostavy za pomoci formulárov (3, s. 29).



**Obrázok 3: Zložky databázového systému**

(Zdroj: Vlastné spracovanie podľa 3, s. 30)

### 1.3. Dátové modely

Pri tvorbe informačného systému si nevystačíme s jedinou štruktúrou vety, preto potrebujeme navrhnuť dátovú štruktúru vety pre každý typ dátového objektu, ktorý je súčasťou informačného systému. Entity spolu v realite súvisia a preto je potrebné vytvoriť, čo najpodrobnejší obraz reality. Pre príklad jednoduchého vytvorenia informačného systému vysokej školy je nutné navrhnuť dátové štruktúry minimálne pre objekty študent, predmet, skúška a učiteľ.

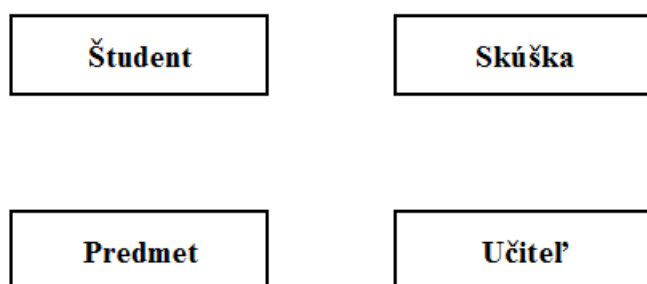


Na vytvorenie informačného systému je možné použiť päť rôznych dátových modelov:

- Lineárny
- Hierarchický
- Sieťový
- Relačný
- Objektový (1, s. 20).

### 1.3.1. Lineárny dátový model

V lineárnom dátovom modeli neexistuje žiadna väzba medzi jednotlivými tabuľkami, preto nie je možné určiť, ktorý študent zložil akú skúšku z akého predmetu (1, s. 20). Tento model je znázornený na obrázku č. 4.



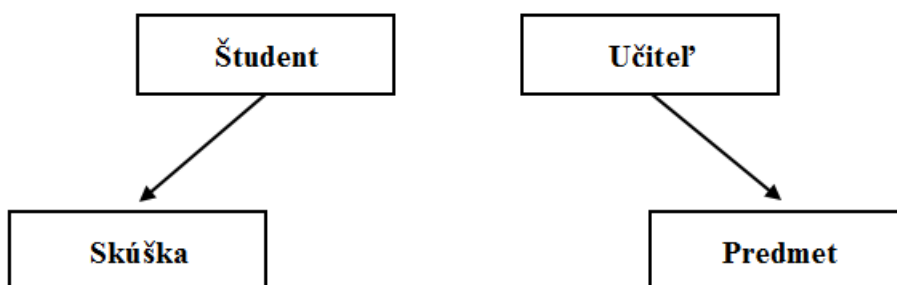
**Obrázok 4: Lineárny dátový model**

(Zdroj: Vlastné spracovanie podľa 1, s. 20)

Príkladom lineárneho modelu môže byť kartotéka pacientov u doktora, kde sú karty s údajmi zoradené v zásuvkách. Každá karta predstavuje jednu vetu databázového súboru (1, s. 20).

### 1.3.2. Hierarchický dátový model

Tento model, ktorý je znázornený aj na obrázku č. 5, je tvorený rodičovským segmentom, z ktorého vedú väzby na jemu podriadené segmenty – čo sú segmenty s inou štruktúrou a obsahom. Väzby medzi jednotlivými segmentami sú vedené takzvanými pointerami, ktoré vytvárajú databázový systém. Na podriadené segmenty je možné sa dostať iba prostredníctvom ich rodičovských segmentov, z toho vyplýva, že nemôžeme čítať údaje o výsledkoch skúšok všetkých študentov. Medzi výhody modelu patrí rýchlosť vyhľadávania pomocou pointerov a prehľadnosť (1, s. 21).

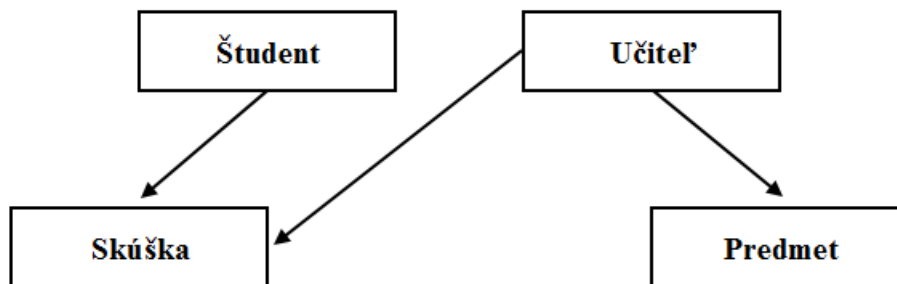


Obrázok 5: Hierarchický dátový model

(Zdroj: Vlastné spracovanie podľa 1, s. 21)

### 1.3.3. Sieťový dátový model

Sieťový model je podobný hierarchickému, od ktorého sa líši tým, že pointerové väzby vedú nielen z hlavného segmentu na vedľajšie, ale obecné v rôznych smeroch medzi databázovými segmentmi. Tieto pointerové väzby znázorňujú kade sú vedené väzby medzi vetami. Výhodou sieťového modelu je ľubovoľné prepájanie segmentov a rýchly prístup k dátam. Model je možné vidieť na obrázku č. 6 (1, s. 22).



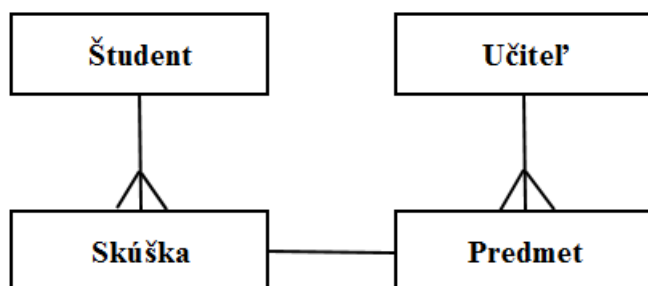
**Obrázok 6: Sieťový dátový model**

(Zdroj: Vlastné spracovanie podľa 1, s. 22)

### 1.3.4. Relačný dátový model

Tento dátový model patrí medzi najpoužívanejšie dátové modely v dnešnej dobe. Pozostáva z niekoľkých lineárnych modelov prepojených pomocou relačného kľúča. Toto spojenie nie je trvalé, ale vzniká v momente, kedy potrebujeme mať dáta zo všetkých spojených tabuliek databázy a zaniká v okamihu ukončenia práce s dátovým modelom (1, s. 22).

Dáta sú v relačnom modeli reprezentované pomocou dvojrozmerných tabuliek. Model taktiež povoľuje zlučovanie tabuliek do pohľadov (4, s. 28). Na obrázku č. 7 môžeme vidieť relačný dátový model, v ktorom čiary na obrázku znázorňujú vzťahy medzi tabuľkami.

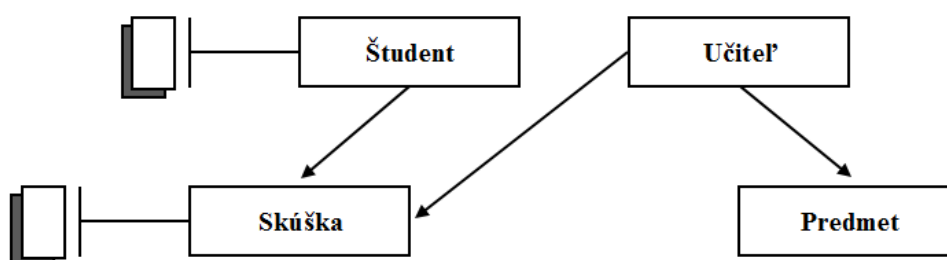


**Obrázok 7: Relačný dátový model**

(Zdroj: Vlastné spracovanie podľa 1, s. 22)

### 1.3.5. Objektový dátový model

Najnovším používaným modelom je objektový dátový model, ktorý má základy postavené na objekte s vlastnými atribútmi a definovanými metódami určujúcimi chovanie objektu. Ak je skúška objektom, atribútmi sú údaje dátum, známka, číslo študenta, termín, predmet a skúšajúci. Tento objekt môže mať definované svoje metódy (1, s. 23). Na obrázku č. 8 je znázornený objektový dátový model.



Obrázok 8: Objektový dátový model

(Zdroj: Vlastné spracovanie podľa 1, s. 23)

## 1.4. Relačný dátový model

Tieto dátové modely sú založené na teórii relácií. Môžeme povedať, že nám umožňujú zachytiť nielen dáta skúmaných objektov, ale aj ich vzťahy, čím sa viac približujú k reálnemu svetu (1, s. 24).

### 1.4.1. Entity

Entitami nazývame reálne objekty, ktorých vlastnosti sledujeme a ukladáme do databázy. Môže sa jednať o človeka, stroj alebo aj zviera, v našom prípade (obrázok č. 7) sú entitami Študent, Skúška, Predmet a Učiteľ (1, s. 12).

### 1.4.2. Atribúty

Pre každú entitu uchovávame ich vlastnosti, ktoré nazývame atribútmi. Atribút vždy popisuje alebo charakterizuje danú entitu (4, s. 41). Pre entitu Študenta môžeme pokladať za atribúty číslo študenta, meno, priezvisko, rodné číslo, adresa...

### 1.4.3. Kľúče

**Kandidátny kľúč** jedinečne identifikuje každý záznam v relácii, môže sa jednať o jednoduchý alebo zložený kľúč. V relácii môžeme nájsť viacero kandidátnych kľúčov, z ktorých zvolíme jeden primárny kľúč.

**Primárny kľúč** je zvolený z kandidátnych kľúčov, umožňuje identifikovať všetky riadky v relácii. Kľúč musí byť jednoznačný a minimálny – nemôžeme z neho odobrať atribút bez porušenia jednoznačnosti (3, s. 82).

**Cudzí kľúč** je primárny kľúč z inej tabuľky, ktorý reprezentuje vzťah medzi týmito tabuľkami (3, s. 86).

### 1.4.4. Relácie

Databázy ukladajú dáta vo forme relácií, ktoré sú konkrétne definované ako dvojrozmerné tabuľky zložené z riadkov a stĺpcov. Tabuľky majú nasledujúce vlastnosti:

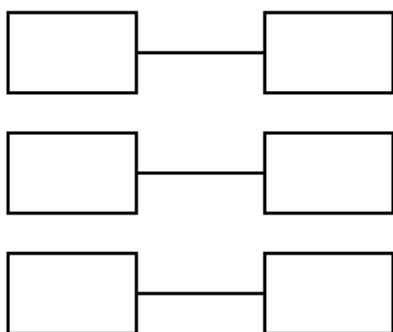
- Riadok obsahuje dáta týkajúce sa danej entity alebo jej časti.
- Stĺpec obsahuje dáta reprezentujúce atribút entity.
- Bunky musia uchovávať jedinú hodnotu – nesmú obsahovať opakujúce sa prvky.
- Všetky položky v stĺpci musia byť rovnakého druhu.
- Názvy stĺpcov musia byť jedinečné.
- Poradie stĺpcov a riadkov nie je dôležité.

- Hodnoty v riadkoch musia byť jedinečné, tabuľka nemôže obsahovať duplicitné dáta (3, s. 78).

### 1.4.5. Integritné obmedzenie pre vzťahy medzi reláciami

Multiplicita obmedzuje počet výskytov entít, ktoré prostredníctvom príslušnej relácie súvisia s inou entitou. Integritné obmedzenie vzťahov obmedzuje ich kardinalitu na pomery 1:1, 1:N, N:1 a N:M (5, s. 163).

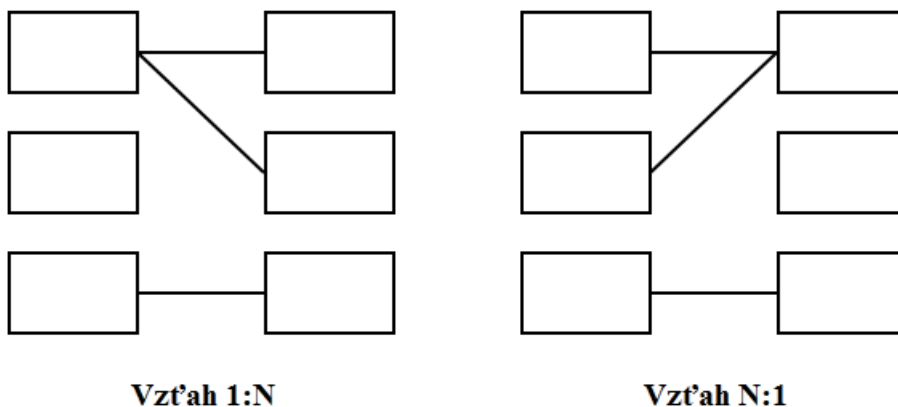
**Vzťah 1:1** znamená, že jednému riadku z prvej tabuľky zodpovedá práve jeden riadok z tabuľky druhej (6, s. 23). Príkladom tejto relácie môže byť vzťah medzi človekom a občianskym preukazom. Jeden človek buď má práve jeden občiansky preukaz alebo nemá žiaden (1, str. 32). Vzťah je zobrazený na nasledujúcom obrázku.



**Obrázok 9: Vzťah 1:1**

(Zdroj: Vlastné spracovanie podľa 6, s. 23)

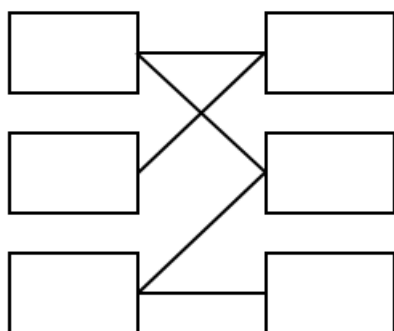
**Vzťah 1:N alebo N:1** znamená, že jednému riadku z prvej tabuľky zodpovedá jeden alebo viac riadkov v tabuľke druhej. Tento typ relácie je v praxi najpoužívanejší (6, s. 23). Ako príklad môže slúžiť vzťah medzi študentom a skúškou. Študent môže vykonať viac skúšok a jedna konkrétna skúška patrí konkrétnemu študentovi (1, s. 33). Vzťahy 1:N a N:1 sú zobrazené na nasledujúcom obrázku.



**Obrázok 10: Vzťah 1:N a N:1**

(Zdroj: Vlastné spracovanie podľa 6, s. 23)

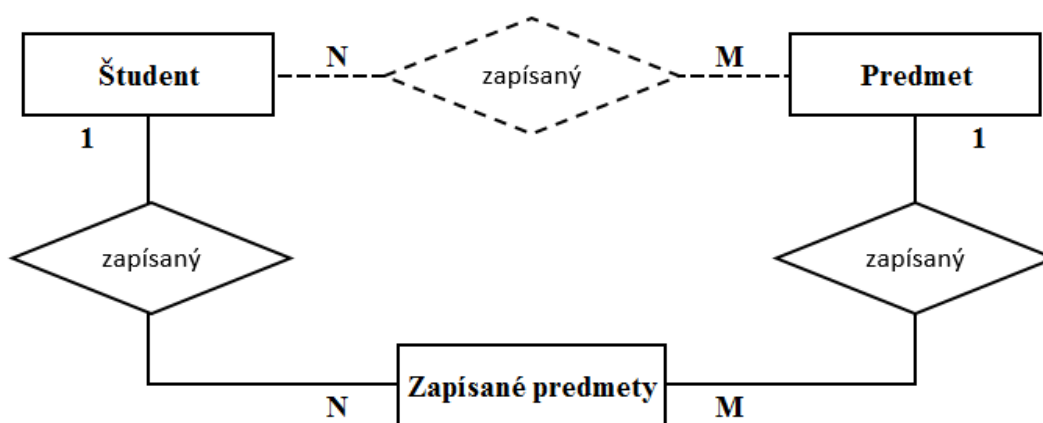
**Vzťah N:M** znamená, že každému riadku prvej tabuľky je možné priradiť jeden alebo viac riadkov v druhej tabuľke a naopak, riadkom druhej tabuľky môže zodpovedať jeden alebo viacero riadkov z prvej tabuľky (6, s. 23). Príkladom je vzťah študenta a predmetu, kde si jeden študent zapisuje viac predmetov a jeden predmet má súčasne zapísaných viac študentov (1, str. 34). Na nasledujúcom obrázku je tento vzťah znázornený.



**Obrázok 11: Vzťah N:M**

(Zdroj: Vlastné spracovanie podľa 6, s. 23)

Pri vzťahu N:M môžeme stanoviť kandidátne kľúče a z nich vybrať primárny kľúč, ale v prípade, že chceme vytvoriť väzbu medzi entitami nastáva problém. Riešenie väzby N:M spočíva v tzv. dekompozícií - vytvorením novej entity s primárnym kľúčom zloženého z primárnych kľúčov pôvodných tabuliek (1, s. 35). Toto riešenie väzby N:M je zobrazené na obrázku č. 12.



**Obrázok 12: Dekompozícia vzťahu N:M**

(Zdroj: Vlastné spracovanie podľa 1, s. 35)

## 1.5. Normálne formy

Pri návrhu relačnej schémy dodržiavame určité pravidlá, ktoré sa nazývajú normálne normy. Správnou normalizáciou dosiahneme odstránenie redundancií pri zachovaní integrity a konzistencie, a vo veľkej miere zefektívňime prácu s databázovými tabuľkami. Čím sú tabuľky relačnej databázy vo vyšších normálnych formách, tým lepšie by sa s nimi malo pracovať z pohľadu aplikačnej logiky (7, s. 168).

### 1.5.1. 1. normálna forma (multizávislosť)

„Tabuľka splňuje podmienku prvej normálnej formy vtedy, ak všetky atribúty (stĺpce) sú atómové, to znamená ďalej nedeliteľné.“ Žiaden stĺpec neobsahuje viac druhov údajov a hodnotou v stĺpci nesmie byť relácia (7, s. 168).



**Tabuľka 1: Tabuľka porušujúca 1. normálnu formu**

(Zdroj: Vlastné spracovanie podľa 1, s. 57)

Meno	Telefónne číslo
Ján Novák	0903524165, 0902547846
Mária Tichá	0914236878
Peter Nagy	0903548965, 0902154689

Príkladom môže byť meno alebo telefónne číslo v **tabuľke č. 1**, ktorú upravíme do 1. normálnej formy (viď **tabuľka č. 2**).

**Tabuľka 2: 1. normálna forma**

(Zdroj: Vlastné spracovanie podľa 1, s. 58)

Meno	Priezvisko	Telefónne číslo 1	Telefónne číslo 2
Ján	Novák	0903524165	0902547846
Mária	Tichá	0914236878	
Peter	Nagy	0903548965	0902154689

**1.5.2. 2. normálna forma (funkčná závislosť)**

*„Tabuľka splňuje podmienku pre zaradenie do druhej normálovej formy vtedy, keď splňuje podmienku prvej normálnej formy a každý atribút okrem primárneho kľúča je úplne závislý na celom primárnom kľúči.“ (7, s. 169).*

Nasledujúca tabuľka obsahuje stĺpce Meno a Priezvisko, ktoré nie sú závislé od celého primárneho kľúča, ale len od stĺpca s názvom Číslo študenta. Môžeme to vyriešiť dekompozíciou tabuľky na dve tabuľky, ktoré neobsahujú neklúčové atribúty závislé na primárnom kľúči.

**Tabuľka 3: Tabuľka porušujúca 2. normálnu formu**

(Zdroj: Vlastné spracovanie)

Číslo študenta	ID Predmetu	Meno	Priezvisko
101545	MA_1	Ján	Novák
101545	INF_1	Ján	Novák
101550	MA_1	Peter	Nagy
101557	INF_1	Mária	Tichá

Výsledok dekompozície tabuľky č. 3, ktorá porušuje 2. normálnu formu, môžeme vidieť v tabuľke č. 4 a tabuľke č. 5.

**Tabuľka 4: 1. tabuľka dekompozície**

(Zdroj: Vlastné spracovanie)

Číslo študenta	Meno	Priezvisko
101545	Ján	Novák
101557	Mária	Tichá
101550	Peter	Nagy

**Tabuľka 5: 2. tabuľka dekompozície**

(Zdroj: Vlastné spracovanie)

Číslo študenta	ID Predmetu
101545	MA_1
101545	INF_1
101550	MA_1

101557	INF_1
--------	-------

### 1.5.3. 3. normálna forma (tranzitívna závislosť)

„Tabuľka je v tretej normálnej forme vtedy, keď je v druhej normálnej forme a zároveň neexistujú závislosti neklúčových stĺpcov tabuľky.“ Ako príklad slúži tabuľka č. 6, ktorá spĺňa predchádzajúce normálne formy a neklúčové stĺpce PSČ a Mesto sú medzi sebou závislé. PSČ by bolo možné vyhľadávať pre dané mesto z tabuliek poštových smerovacích čísiel, ak by to bolo výhodnejšie (7, str. 170).

**Tabuľka 6: Tabuľka porušujúca 3. normálnu formu**

(Zdroj: Vlastné spracovanie)

Priezvisko	Mesto	PSČ
Novák	Košice	04001
Tichá	Bratislava	84101
Nagy	Žilina	01001

V nasledujúcich dvoch tabuľkách (tabuľka č.7 a tabuľka č.8) je znázornený výsledok dekompozície, ktorý spĺňa 3. normálnu formu.

**Tabuľka 7: 1. tabuľka dekompozície**

(Zdroj: Vlastné spracovanie)

Priezvisko	Mesto
Novák	Košice
Tichá	Bratislava
Nagy	Žilina

**Tabuľka 8: 2. tabuľka dekompozície**

(Zdroj: Vlastné spracovanie)

<b>Mesto</b>	<b>PSČ</b>
Košice	04001
Bratislava	84101
Žilina	01001

## **1.6. Návrh databázy**

Návrh databázy pozostáva z troch po sebe nasledujúcich fáz. Začína sa fázou konceptuálneho návrhu, pokračuje logický návrh a konečnou fázou je fyzický návrh. Štruktúrovaný prístup podporujúci a zjednodušujúci proces návrhu, ktorý používa procedúry, techniky, nástroje a dokumentáciu, sa nazýva metodológia návrhu (5, s. 206).

### **1.6.1. Konceptuálny návrh**

Prvou fázou návrhu databázy je konceptuálny návrh. Je to proces vytvárania modelu na základe firmou používaných dát bez toho, aby sme mysleli na podkladový dátový model (napr. relačný dátový model) alebo fyzickú implementáciu. Identifikuje dôležité entity a relácie potrebné reprezentovať v databáze a je zdrojom informácií pre ďalšiu fázu návrhu – logického návrhu (5, s. 206).

### **1.6.2. Logický návrh**

Logický návrh je druhou fázou návrhu databázy. Pri tejto fáze sa vytvára logická reprezentácia databázy použitím napr. relačného modelu dát, v ktorom dôležité entity a relácie reprezentujeme ako množinu relačných tabuliek. Tento návrh slúži ako zdroj informácií k fyzickému návrhu a nástroj pre zvažovanie alternatívnych postupov pre efektívny návrh databázy (5, s. 206).

### 1.6.3. Fyzický návrh

V poslednej fáze návrhu databázy sa popisujú podkladové tabuľky, organizácie súborov, indexy pre dosiahnutie efektívneho dátového prístupu, všetky integritné a bezpečnostné obmedzenia. Taktiež sa návrhár databázy môže rozhodnúť o spôsobe najvhodnejšej fyzickej implementácie logického návrhu a vhodne upravuje návrh pre konkrétny databázový systém ako je napr. MS Access, SQL Server alebo Oracle (5, s. 207).

## 1.7. Jazyk SQL

Vyhľadávací jazyk SQL používame pri komunikácii s relačnými databázami. Na rozdiel od procedurálnych programovacích jazykov ako je C, C++, C# alebo Java, SQL patrí medzi neprocedurálne programovacie jazyky, ktorými popisujeme čo od databázy požadujeme, nie ako to treba urobiť. Môžeme v jazyku SQL definovať dáta, vykonávať aktualizácie aj nastavovať prístupové práva k databáze. Modelovanie dát pomocou SQL má tieto rysy:

- Dáta môžu byť skutočné alebo virtuálne a sú uložené vo forme tabuliek (8, s. 57).
- Dáta vracia programu (alebo užívateľovi), ktorý sa nemusí starať o umiestnenie týchto dát a ich fyzickú štruktúru.
- Tabuľky sú identifikované menom a ich poloha v databáze nie je dôležitá.
- Na poradí stĺpcov v tabuľkách nezáleží, sú identifikované menom.
- Poradie riadkov v tabuľkách nie je dôležité, identifikujú sa hodnotami v stĺpcoch.
- Bez ohľadu na vnútornú štruktúru dát použitú v databáze, sú dáta vždy prezentované užívateľovi ako tabuľky (8, s. 58).

### **1.7.1. Kategórie príkazov**

SQL príkazy sa delia do kategórií podľa svojej funkcie. Všetky kategórie majú rovnakú základnú syntax a pravidlá, z toho dôvodu ich môžeme považovať za kategórie príkazov v rámci jedného jazyka. Poznáme päť kategórií jazyka SQL:

- Jazyk DDL,
- Jazyk DQL,
- Jazyk DML,
- Jazyk DCL,
- Príkazy riadenia transakcií (9, s. 40).

### **1.7.2. Jazyk DDL**

Jazyk pre definíciu dát alebo jazyk DDL podporuje vytváranie, odstraňovanie, upravovanie definícií a štruktúr tabuliek a pohľadov (8, s. 59).

Súčasťou DDL sú príkazy CREATE, ALTER a DROP, ktoré majú vplyv na databázové tabuľky s dátami a nie na uložené dáta. Tieto príkazy neumožňujú vytvárať či meniť riadky dát v tabuľkách. Príkazom CREATE je možné vytvoriť databázu alebo tabuľku, naopak príkaz DROP zmaže už existujúcu tabuľku. Posledným príkazom je ALTER, vďaka ktorému je možné existujúce tabuľky upravovať (10, s. 342).

### **1.7.3. Jazyk DQL**

Jazyk DQL je veľmi dôležitá súčasť jazyka SQL. Základným príkazom je príkaz SELECT, ktorý sa používa na načítavanie dát z jednej alebo viacerých tabuliek databázy (9, s. 41).

Príkaz SELECT je najpoužívanejší príkaz SQL jazyka, ktorý okrem vyhľadávania dát je schopný ich aj vhodne zoradiť, vytvárať vypočítané polia a súhrny. Vždy obsahuje tri základné časti:

- Klauzulu SELECT
- Zoznam všetkých polí, ktoré chceme zobrazit'. Polia oddeľujeme čiarkami a symbolom hviezdičky zobrazí všetky polia.
- Klauzulu FROM, ktorá je nasledovaná názvom tabuľky (11, s. 101).

#### **1.7.4. Jazyk DML**

Táto súčasť jazyka SQL obsahuje príkazy, ktoré umožňujú pridávať odoberať alebo meniť stávajúce dáta v databáze. Medzi tri základné príkazy DML patria príkazy INSERT, UPDATE a DELETE (9, s. 41).

Príkaz INSERT pridáva nové riadky dát do databázovej tabuľky, UPDATE aktualizuje respektíve upravuje existujúce dáta v tabuľke a DELETE z tabuľky odstraňuje riadky. Tieto príkazy môžu manipulovať len s dátami jednej tabuľky (11, s. 257).

#### **1.7.5. Jazyk DCL**

Do tejto kategórie patria príkazy, ktoré dovoľujú správcovi databázy riadiť prístup k dátam a používať rôzne systémové oprávnenia. Príkladom je funkcia pre spustenie alebo vypnutie databázy. Medzi základné príkazy patria príkazy GRANT, REVOKE a ALTER (9, s. 41).

Užívateľom databázy je možné udeliť rôzne oprávnenia pomocou príkazu GRANT a naopak príkazom REVOKE je možné tieto oprávnenia odobrať (9, s. 150).

#### **1.7.6. Príkazy riadenia transakcií**

Databázová transakcia je súbor príkazov, ktorú užívateľ databázy požaduje spracovať ako nedeliteľný celok, čo znamená, že transakcia musí byť kompletne úspešná alebo neúspešná. Príkazy nezodpovedajú syntaxi príkazov jazyka SQL, ale majú veľký vplyv na chovanie príkazov, ktoré sú súčasťou danej transakcie (9, s. 41).

## **2 ANALÝZA SÚČASNÉHO STAVU**

V nasledujúcej kapitole zanalyzujem súčasný stav jazykovej školy, pre ktorú bude daný systém implementovaný. Najskôr spíšem základné informácie o škole a ich organizačnej štruktúre. Ďalej sa budem venovať analýze aktuálneho stavu spracovávaní údajov.

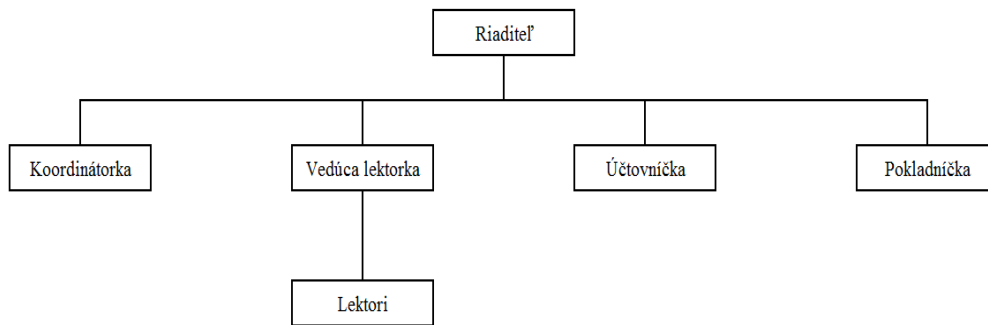
### **2.1. Základné informácie o jazykovej škole**

Jazyková škola bola založená v roku 2015. Jazyky vyučuje na štyroch základných úrovniach: začiatočník, mierne pokročilý, stredne pokročilý a pokročilý. V ponuke má päť svetových jazykov. Konkrétne sa jedná o jazyk anglický, nemecký, španielsky, francúzsky a ruský. Verejnosti poskytujú širokú ponuku skupinových kurzov, kde sa počet študentov v triedach pohybuje v rozmedzí 4 až 6 vo väčších skupinách alebo v menších skupinkách s 2 až 3 študentmi. Pre náročnejších poskytujú aj individuálne kurzy, konverzácie alebo prípravu na anglické jazykové skúšky PET, FCE a CAE. Rodiny s malými deťmi od 2 do 4 rokov sa môžu zapísať na hodiny angličtiny. Ďalej umožňujú firmám dohodnúť si kurz na mieru presne podľa ich potrieb či už v ich firme alebo v priestoroch jazykovej školy a popíšem procesy prihlasovania a evidovania študentov.

### **2.2. Organizačná štruktúra**

V škole pracuje 14 zamestnancov na plný úväzok. Školu zastupujú 2 jednatelia, z toho jeden je zároveň riaditeľom školy. Riaditeľ jazykovej školy má na starosti víziu a rozvoj školy, vylepšuje a rozvíja existujúce služby a kurzy a taktiež má na starosti budovu školy. O finančnú stránku školy sa stará účtovníčka spolu s pokladničkou. Koordinátorka sa zaoberá prepojením študentov s lektormi a školou, ale aj lektorov so školou. Hlavným zdrojom príjmu je 8 jazykových lektorov, ktorých má na starosti vedúca lektorka zastávajúca zároveň aj funkciu metodičky výukovej stratégie.





**Obrázok 13: Organizačná štruktúra jazykovej školy**

(Zdroj: vlastné spracovanie)

### 2.3. Popis súčasnej situácie

Škola momentálne nemá ucelený systém pomocou ktorého by mohla jednoducho spracovávať a ukladať všetky evidované údaje svojich zamestnancov, študentov a ponúkaných kurzov.

Na uchovávanie základných údajov o študentoch a zamestnancoch používajú v škole online Google tabuľky, v ktorých sú údaje uložené nezávisle v dvoch samostatných tabuľkách. Takisto tu môžeme nájsť informácie kurzoch, ale tie sú len minimálne a spočívajú hlavne v jednej tabuľke obsahujúcej záznamy o tom, ktorý lektor má na starosti výučbu konkrétneho kurzu.

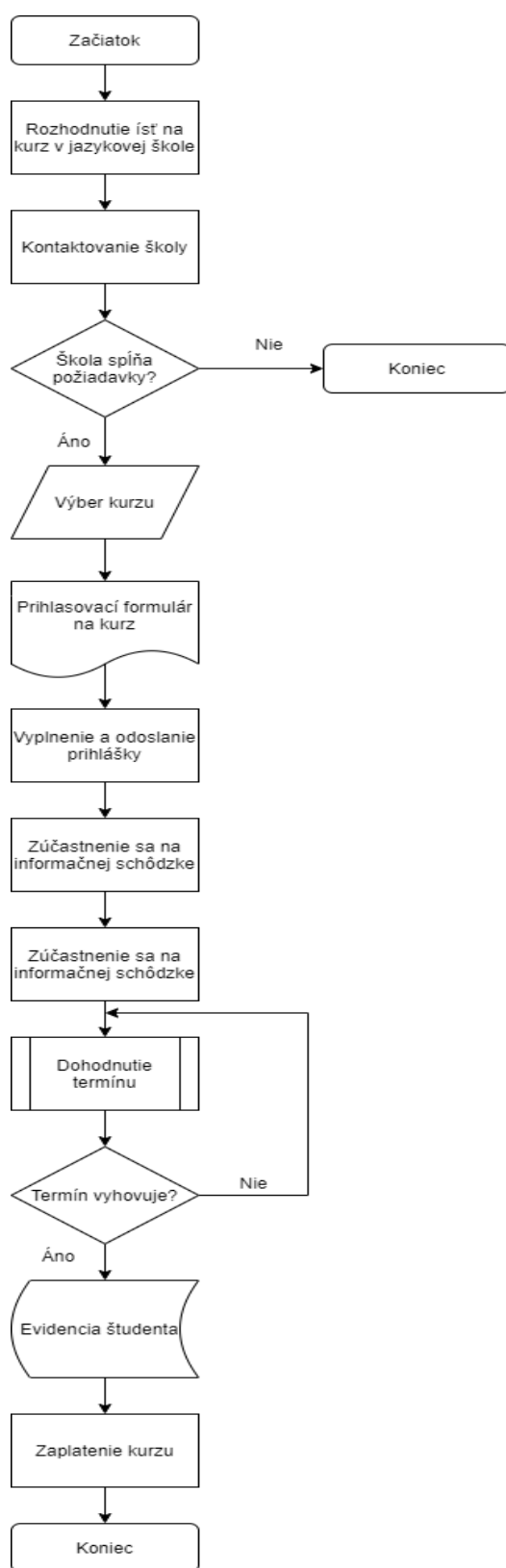
Tieto používané tabuľky nie sú nijakým spôsobom prepojené ani automatizované, čo vedie k neefektívnej práci s dátami. Riešením tohto problému by pomohol jednotný databázový systém, ktorý by obsahoval všetky požadované informácie. Nový systém by prepojil všetky aktuálne komponenty a tým zjednodušil, zrýchlil a zefektívnil spracovávanie týchto dát. Veľký vplyv by mal aj na vyhľadávanie konkrétnych informácií v administratíve školy, ktoré je v súčasných tabuľkách zložitejšie a časovo náročnejšie.

## 2.4. Prihlasovanie študentov

V súčasnosti sa na kurz prihlasuje pomocou elektronického formulára na webových stránkach školy, osobne v budove školy alebo v prípade študentov, ktorí už na škole študovali predchádzajúci semester prostredníctvom emailu obsahujúcim informáciou o záujme v pokračovaní štúdia.

Proces prihlasovania na kurz začína rozhodnutím študovať na škole, následnom kontaktovaní školy za účelom získania ďalších informácií a ak škola vyhovuje študentovým požiadavkám, študent pokračuje výberom kurzu. Po nájdení vhodného kurzu, študent obdrží od školy prihlasovací formulár, ktorý vyplní. V prihláške študenti vypisujú základné údaje ako je meno, priezvisko a veková kategória študenta, následne kontaktné údaje ako je email a telefón. Nakoniec si študent zvolí jazyk, ktorý chce študovať, približnú úroveň zdatnosti vo vybranom jazyku a aký typ kurzu chce študovať. Detaily ohľadom termínu kurzu sa dohadujú osobne na informačnej schôdzke so študentmi alebo pomocou elektronickej komunikácie. Medzi posledné kroky prihlásenia patrí evidencia študenta do školských záznamov a zaplatenie školného.

Celý proces prihlásenia študenta od prvého rozhodnutia až po záväzné prihlásenie na kurz je zobrazený vývojovým diagramom na nasledujúcom obrázku.

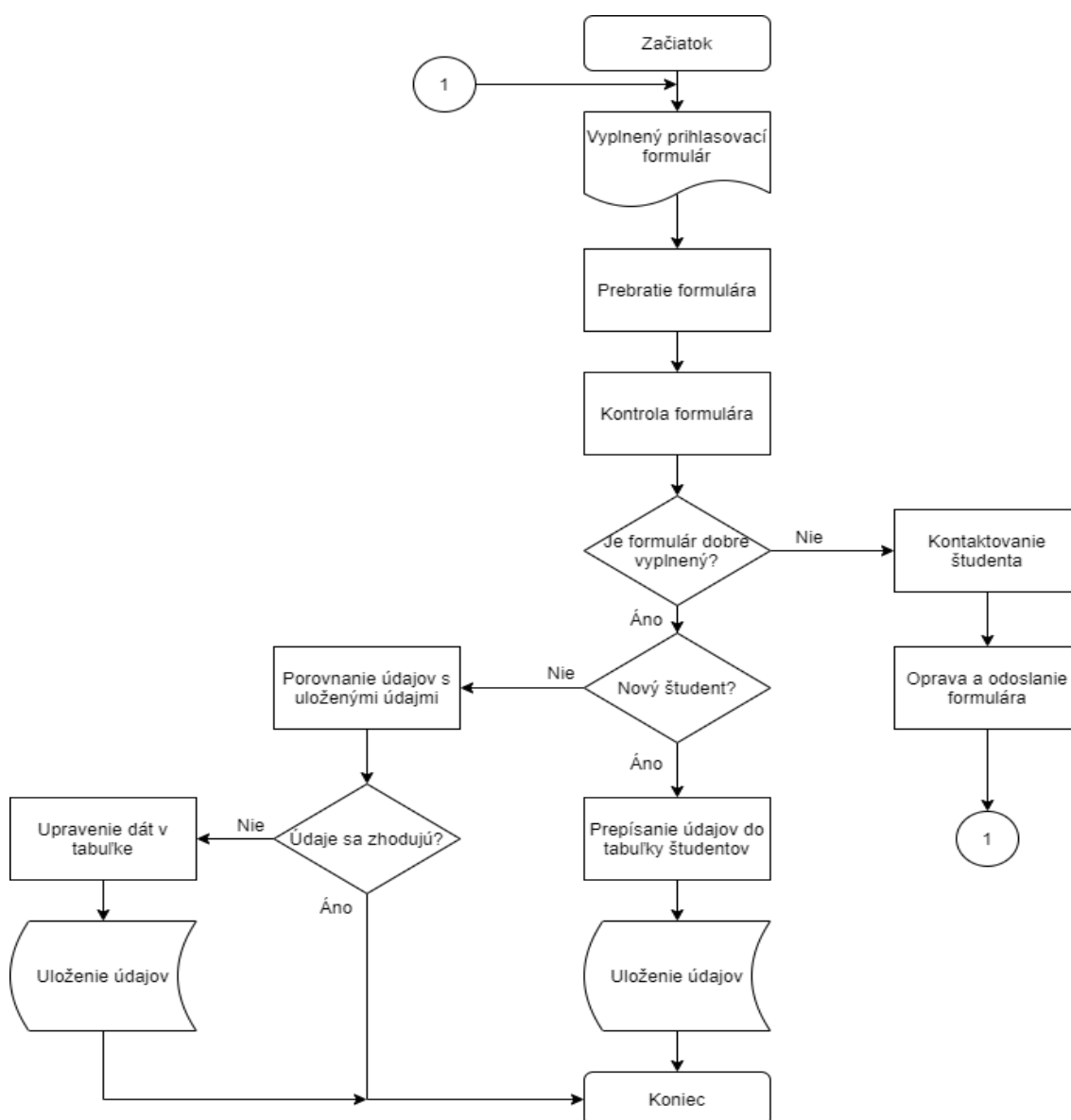


**Obrázok 14: Vývojový diagram prihlásenia na kurz**

(Zdroj: Vlastné spracovanie)

## 2.5. Evidencia študentov

Po vyplnení prihlášky a dohodnutí na termíne kurzu nasleduje evidencia študenta. Najskôr sa kontroluje úplnosť vyplnenia formulára. V prípade, že nejaký údaj chýba, je nutné formulár poslať študentovi na doplnenie. Ak je všetko v poriadku vyplnené a študent ešte nebol v škole zapísaný, tak sa všetky jeho základné údaje prepíšu do tabuľky študentov a následne uložia. Ak ide o študenta, ktorý už školu navštevoval, tak sa iba prekontrolujú uložené údaje a v prípade zmien sa neaktuálne údaje prepíšu novými a uložia. Tento proces je zobrazený vývojovým diagramom na obrázku č. 15.



Obrázok 15: Vývojový diagram evidencie študentov

(Zdroj: Vlastné spracovanie)

## 3 VLASTNÉ NÁVRHY RIEŠENIA

Táto kapitola obsahuje riešenie nedostatkov jazykovej školy, ktoré sú v popísané v predchádzajúcej kapitole. Pre lepší prehľad je rozdelená do troch hlavných častí, kde prvá časť je venovaná konceptuálnemu návrhu, nasledujúca logickému návrhu a v poslednej je popísaný fyzický návrh. Obsahom je návrh databázy spĺňajúci podmienky jazykovej školy a je vhodným prostriedkom pre organizáciu a správu dát nutných k jej lepšiemu fungovaniu.

### 3.1. Konceptuálny návrh

Nasledujúca kapitola sa zaoberá údajmi z reálneho sveta, s ktorými bude databáza pracovať. Na základe týchto údajov sú definované entity a ich vzájomné vzťahy. Informácie z tejto kapitoly budú ďalej slúžiť k zostavovaniu jednotlivých tabuliek a ich atribútov.

#### 3.1.1. Identifikácia entít

V nasledujúcej tabuľke sú znázornené dôležité entity spojené s návrhom databázy, ich stručné popisy a množstvo ich výskytu v databáze.

**Tabuľka 9: Identifikácia entít**

(Zdroj: Vlastné spracovanie)

Entita	Popis	Výskyt
Zamestnanec	Všetci zamestnanci školy	Desiatky
Študent	Všetci študenti školy	Stovky
Kurz	Všetky kurzy školy	Desaťtisíce
Úroveň	Úrovne pokročilosti jazyka	Jednotky
Miestnosť	Priestory na učenie v škole	Desiatky

Typ kurzu	Vyučované typy kurzov	Jednotky
Jazyk	Vyučované jazyky	Jednotky
Pozícia	Pozície zamestnancov	Jednotky
Veková kategória	Veková kategória prihlasovaných študentov	Jednotky
Deň	Vyučované dni	Jednotky
PSČ	Zoznam miest podľa PSČ	Desaťtisíce

### 3.1.2. Identifikácia relácií

Táto podkapitola obsahuje tabuľku so základnými vzťahmi medzi entitami identifikovanými v predchádzajúcej kapitole a kardinalitu týchto vzťahov.

**Tabuľka 10: Identifikácia vzťahov**

(Zdroj: Vlastné spracovanie)

Entita	Relácia	Entita	Kardinalita
Zamestnanec	učí	Študenta	N:M
Zamestnanec	učí	Jazyk	N:M
Zamestnanec	má	Pozíciu	N:M
Zamestnanec	vyučuje	Kurz	N:M
Zamestnanec	býva	PSČ	N:1
Študent	má zapísaný	Kurz	N:M
Študent	je vo	Vekovej kategórii	N:1
Študent	ovláda	Jazyk	N:M
Kurz	je	Typ kurzu	N:1

Kurz	je vyučovaný v	Jazyku	N:1
Kurz	je vyučovaný v	Úrovni	N:1
Kurz	je vyučovaný v	Deň	N:1
Kurz	je vyučovaný v	Miestnosti	N:1

## 3.2. Logický návrh

Obsahom kapitoly logického návrhu sú definované jednotlivé tabuľky, ktoré budú obsiahnuté vo výslednej databáze. Každá tabuľka pozostáva z názvu, popisu a dátového typu atribútu. Primárne kľúče sú vyznačené modrou farbou a cudzie kľúče šedou farbou.

### 3.2.1. Tabuľka Zamestnanec

V priloženej tabuľke sú evidované všetky osobné údaje o zamestnancovi, ktoré je potrebné uchovávať.

**Tabuľka 11: Atribúty tabuľky Zamestnanec**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo zamestnanca (automaticky generované)	Int	Nie
Meno	Meno zamestnanca	Varchar 20	Nie
Priezvisko	Priezvisko zamestnanca	Varchar 30	Nie
Email	Email zamestnanca	Varchar 50	Ano

Telefon	Telefónne číslo na zamestnanca	Varchar 12	Nie
PSC	PSC bydliska (prepojenie s tabuľkou PSČ)	Varchar 6	Nie
Ulica	Ulica bydliska zamestnanca	Varchar 20	Nie
Popisne_c	Popisné číslo bydliska zamestnanca	Varchar 10	Nie
Datum_nar	Dátum narodenia zamestnanca	Date	Nie
Cislo_uctu	Číslo bankového účtu	Varchar 20	Nie

### 3.2.2. Tabuľka Študent

Tabuľka uchováva všetky potrebné osobné a kontaktné údaje o študentovi zapísaného do jazykového kurzu.

**Tabuľka 12: Atribúty tabuľky Študent**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo študenta (automaticky generované)	Int	Nie
Meno	Meno študenta	Varchar 20	Nie
Priezvisko	Priezvisko študenta	Varchar 30	Nie
Email	Email študenta	Varchar 50	Nie



Telefón	Telefónne číslo na študenta	Varchar 12	Áno
Vekova_kat	Veková kategória do ktorej študent spadá (prepojenie s tabuľkou Veková kategória)	Int	Nie

### 3.2.3. Tabuľka Kurz

Táto tabuľka eviduje dôležité údaje spojené s vyučovanými kurzmi. Je tu zahrnutý jeho názov, typ, vyučovaný jazyk s informáciou o jeho úrovni a miesto a čas konania kurzu.

**Tabuľka 13: Atribúty tabuľky Kurz**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo kurzu (automaticky generované)	Int	Nie
Nazov	Názov kurzu	Varchar 50	Nie
Typ_kurzu	Typ kurzu (prepojenie s tabuľkou Typ kurzu)	Int	Nie
Jazyk	Jazyk kurzu (prepojenie s tabuľkou Jazyk)	Int	Nie
Uroven	Jazyková úroveň	Int	Nie

	(prepojenie s tabuľkou Úroveň)		
Vekova_kat	Veková kategória študentov (prepojenie s tabuľkou Veková kategória)	Int	Nie
Miestnost	Miestnosť v ktorej kurz prebieha (prepojenie s tabuľkou Miestnosť)	Int	Nie
Den	Deň v týždni v ktorý kurz prebieha (prepojenie s tabuľkou Deň)	Varchar 3	Nie
Cas	Začiatok kurzu	Time	Nie

### 3.2.4. Tabuľka Typ kurzu

Tu evidujeme či je kurz firemný, skupinový, individuálny alebo minikurz s celkovým časovým určením, kapacitou a cenou daného typu kurzu.

**Tabuľka 14: Atribúty tabuľky Typ kurzu**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo typu kurzu (automaticky generované)	Int	Nie

Nazov	Názov typu kurzu	Varchar 50	Nie
Trvanie	Trvanie kurzu v minútach	Varchar 10	Nie
Cena	Cena kurzu	Int	Nie
Kapacita	Kapacita kurzu	Int	Nie

### 3.2.5. Tabuľka Jazyk

Tabuľka obsahuje údaje o jazyku, ktorý vyučujú na škole.

**Tabuľka 15: Atribúty tabuľky Jazyk**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo jazyka (automaticky generované)	Int	Nie
Nazov	Názov jazyka	Varchar 20	Nie

### 3.2.6. Tabuľka Úroveň

Tabuľka obsahuje údaje o možných úrovniach vyučovaných jazykov a popise vyžadovaných znalostí danej úrovne.

**Tabuľka 16: Atribúty tabuľky Úroveň**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo jazykovej úrovne (automaticky	Int	Nie

	generované)		
Nazov	Názov jazykovej úrovne	Varchar 20	Nie
Popis	Vyžadované znalosti na dosiahnutie úrovne	Varchar 200	Áno

### 3.2.7. Tabuľka Miestnosť

Tabuľka obsahuje údaje o miestnostiach v jazykovej škole a ich účelu.

**Tabuľka 17: Atribúty tabuľky Miestnosť**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo miestnosti (automaticky generované)	Int	Nie
Nazov	Názov miestnosti	Varchar 20	Nie
Popis	Popis miestnosti	Varchar 100	Áno
Kapacita	Celková kapacita	Int	Nie

### 3.2.8. Tabuľka Pozícia

Táto tabuľka eviduje konkrétne názvy pozícií vo firme aj s ich popisom.

**Tabuľka 18: Atribúty tabuľky Pozícia**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo pracovnej pozície (automaticky generované)	Int	Nie
Nazov	Názov pracovnej pozície	Varchar 30	Nie
Popis	Popis pracovnej pozície	Varchar 200	Áno

### 3.2.9. Tabuľka Deň

Tabuľka obsahuje údaje o dňoch v týždni.

**Tabuľka 19: Atribúty tabuľky Deň**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
kod	Trojmiestny kód dňa v týždni	Varchar 3	Nie
Nazov	Názov dňa v týždni	Varchar 10	Nie

### 3.2.10. Tabuľka Veková kategória

Tabuľka obsahuje údaje o vekových kategóriách, podľa ktorých sa študenti rozdeľujú do kurzov.

### Tabuľka 20: Atribúty tabuľky Veková kategória

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo vekovej kategórie (automaticky generované)	Int	Nie
Nazov	Názov vekovej kategórie	Varchar 20	Nie

### 3.2.11. Tabuľka PSČ

Táto tabuľka obsahuje poštové smerovacie čísla s konkrétnymi mestami.

### Tabuľka 21: Atribúty tabuľky PSČ

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo PSČ	Int	Nie
Mesto	Názov mesta	Varchar 25	Nie

### 3.2.12. Dekompozičná tabuľka Zapísaný kurz

Nasledujúca tabuľka slúži ako dekompozičná tabuľka relácie typu N:M medzi tabuľkami Študent a Kurz a Zamestnanec a Študent. Jej obsahom sú údaje o zapísanom kurze študenta a lektora, ktorý ho daný kurz vyučuje.

**Tabuľka 22: Atribúty tabuľky Zapísaný kurz**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo (automaticky generované)	Int	nie
ID_student	Identifikačné číslo študenta (prepojenie s tabuľkou Študent)	Int	Nie
ID_kurz	Identifikačné číslo kurzu (prepojenie s tabuľkou Kurz)	Int	Nie
ID_zamestnanec	Identifikačné číslo lektora (prepojenie s tabuľkou Zamestnanec)	Int	Nie
Poznamky	Poznámky	Varchar 200	Áno
Zaplatene	Informácia o zaplatení kurzu	Bit	Nie

### 3.2.13. Dekompozičná tabuľka Vyučovaný jazyk

Tabuľka je dekompozičnou tabuľkou vzťahu typu N:M medzi tabuľkami Zamestnanec a Jazyk. Zobrazuje informácie o jazykoch, ktoré učí daný lektor na škole.

**Tabuľka 23: Atribúty tabuľky Vyučovaný jazyk**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo (automaticky generované)	Int	Nie
ID_zamestnanec	Identifikačné číslo lektora (prepojenie s tabuľkou Zamestnanec)	Int	Nie
ID_jazyk	Identifikačné číslo jazyka (prepojenie s tabuľkou Jazyk)	Int	Nie
ID_uroven	Identifikačné číslo úrovne (prepojenie s tabuľkou Úroveň)	Int	Nie
poznamky	Poznámky	Varchar 200	Áno

### 3.2.14. Dekompozičná tabuľka Pozícia zamestnanca

V tejto dekompozičnej tabuľke väzby N:M medzi tabuľkami Zamestnanec a Pozícia, je možné nájsť informácie ohľadom toho, akú pozíciu daný zamestnanec zastáva.

**Tabuľka 24: Atribúty tabuľky Pozícia zamestnanca**

(Zdroj: Vlastné spracovanie)

Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo (automaticky	Int	Nie



	generované)		
ID_zamestnanec	Identifikačné číslo zamestnanca (prepojenie s tabuľkou Zamestnanec)	Int	Nie
ID_pozicia	Identifikačné číslo pozície (prepojenie s tabuľkou Pozícia)	Int	Nie

### 3.2.15. Dekompozičná tabuľka Znalosť jazyka

Posledná dekompozičná tabuľka vzťahu typu N:M tabuliek Študent a Jazyk zachytáva informácie ohľadom dosiahnutej študentovej úrovne konkrétneho vyučovaného jazyka.

**Tabuľka 25: Atribúty tabuľky Znalosť jazyka**

(Zdroj: Vlastné spracovanie)

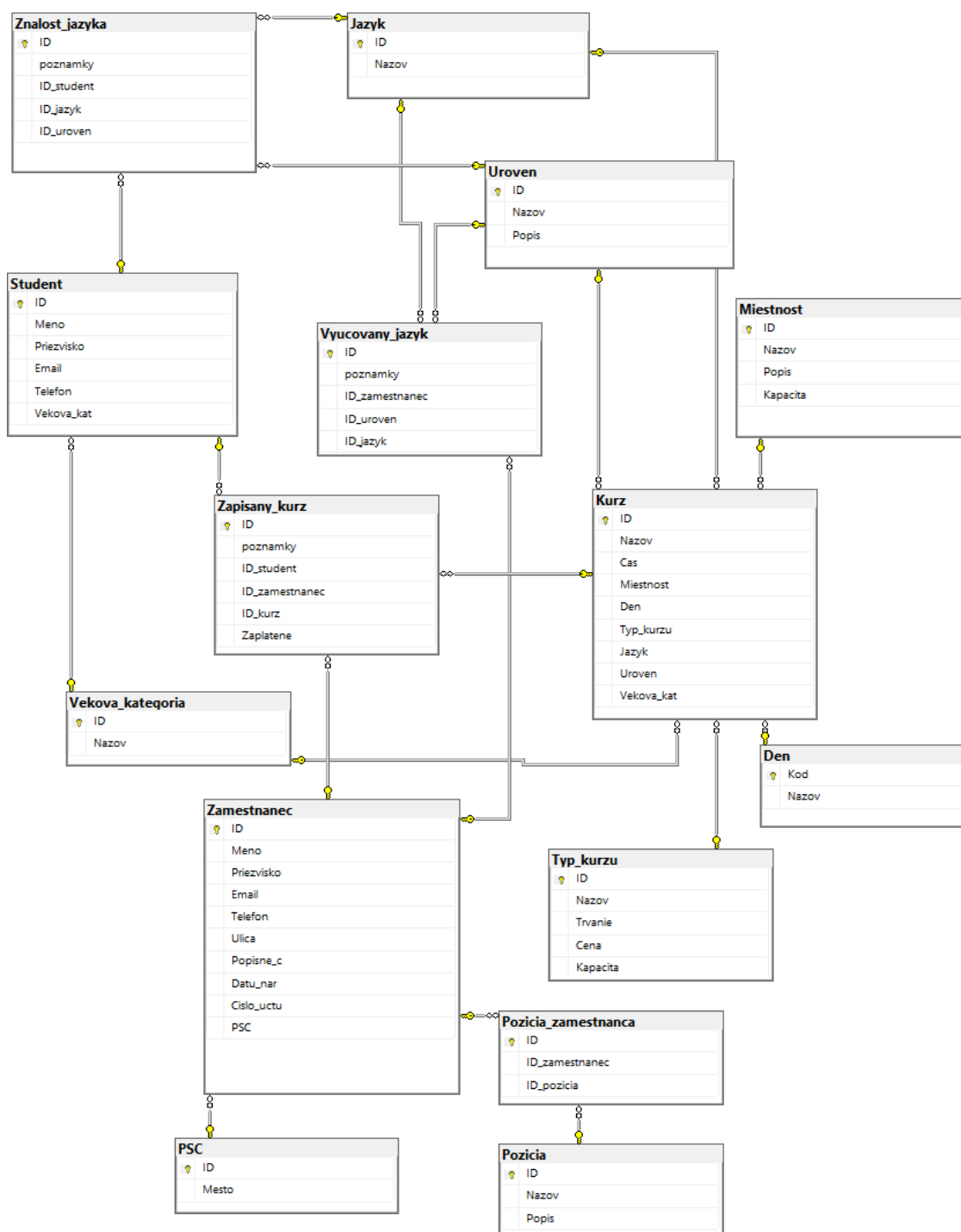
Názov atribútu	Popis	Dátový typ a dĺžka	Null
ID	Identifikačné číslo (automaticky generované)	Int	Nie
ID_student	Identifikačné číslo študenta (prepojenie s tabuľkou Študent)	Int	Nie
ID_jazyk	Identifikačné číslo jazyka (prepojenie s tabuľkou Jazyk)	Int	Nie

ID_uroven	Identifikačné číslo úrovne (prepojenie s tabuľkou Úroveň)	Int	Nie
poznanky	Poznámky	Varchar 200	Áno

### 3.2.16. ER Diagram

Na obrázku č. 16 je zobrazený ER diagram navrhnutého databázového systému. Diagram zahŕňa všetky databázové tabuľky s jednotlivými atribútmi. V tabuľkách sú označené primárne kľúče, ktoré sú znázornené malým obrázkom kľúča na ľavej strane riadka. Zobrazené sú aj jednotlivé väzby medzi týmito tabuľkami. Tabuľky spĺňajú požiadavky všetkých troch normálnych foriem. Pre ich splnenie bolo vytvorených niekoľko číselníkov.

V hlavičkách jednotlivých tabuliek sú uvedené názvy tabuľky, ktoré sú napísané v podobe SQL kódu. Rovnako ako názvy, tak aj atribúty tabuliek sú v tejto podobe.



**Obrázok 16: ER-diagram**

(Zdroj: Vlastné spracovanie)

### 3.3. Fyzický návrh

Táto kapitola sa venuje poslednej fáze návrhu databázy a tou je fyzická návrh, ktorý vychádza z logického návrhu databázy v predchádzajúcej kapitole.

Pozostáva z podkapitol, v ktorých je postupne popísané tvorenie databázy pomocou jazyka SQL. Ako prvý krok je vytvorenie databázy v ktorej budem vytvárať tabuľky a vkladať dáta. Následne vytvorením tabuľky s primárnymi kľúčmi a definovanými integritnými obmedzeniami. Plynule po vytváraní tabuliek nasleduje ich prepájanie prostredníctvom pridelovania cudzích kľúčov a vloženie dát pre testovacie účely. Pre pohodlné zobrazovanie dát v prijateľnej forme pre užívateľov databázy sú v ďalšom kroku vytvorené pohľady. V poslednej podkapitole je vytvorená spúšť mazania študenta.

#### 3.3.1. Vytvorenie databázy

Na začiatku vytvorím vlastnú databázu s vlastným názvom pomocou príkazu CREATE DATABASE. Nižšie je vypísaný kód slúžiaci k vytvoreniu databázy.

Kód:

```
CREATE DATABASE Bakalarska_praca
```

#### 3.3.2. Vytvorenie tabuliek

Po úspešnom vytvorení databázy nasleduje vytvorenie jednotlivých tabuliek popísaných v logickom návrhu. Tabuľky sa vytvárajú príkazom CREATE TABLE. Príkaz obsahuje názvy jednotlivých stĺpcov, ich dátové typy a obmedzenia. Ako obmedzenie stĺpca berieme určenie primárneho kľúča, nulové alebo nenulové atribúty.

Ďalej je zobrazená časť kódu, ktorý vytvorí tabuľku Zamestnanca s atribútmi ID, Meno, Priezvisko, Email, Telefón, Ulica, Popisné číslo, Dátum narodenia, Číslo účtu, PSČ. V prílohe č. 1 je uvedený celý kód pre vytvorenie všetkých tabuliek databázy.

Kód:

```
CREATE TABLE Zamestnanec
(
ID int NOT NULL identity(1,1),
Meno varchar(20) NOT NULL ,
Priezvisko varchar(30) NOT NULL ,
Email varchar(50) ,
Telefon varchar(12) NOT NULL ,
Ulica varchar(20) NOT NULL ,
Popisne_c varchar(10) NOT NULL ,
Datu_nar date NOT NULL ,
Cislo_uctu varchar(20) NOT NULL ,
PSC varchar(6) NOT NULL ,

CONSTRAINT PK_Zamestnanec PRIMARY KEY (ID),
)
GO
```

### 3.3.3. Definovanie cudzích kľúčov

V tomto kroku prepojíme medzi sebou jednotlivé tabuľky definovaním cudzích kľúčov. Kľúče je možné vytvoriť zároveň s vytváraním jednotlivých tabuliek alebo pomocou príkazu ALTER TABLE po vytvorení tabuliek.

V práci je použitý druhý spôsob príkazom ALTER TABLE, ktorý je možné vidieť v prepísanej časti kódu nižšie. Ako príklad je zvolená tabuľka Zamestnanca, o ktorej z ER diagramu vyplýva, že má práve jeden cudzí kľúč, ktorý je primárnym kľúčom v tabuľke PSC. Celý kód obsahujúci definovanie kľúčov je súčasťou prílohy č. 1 na konci práce.

Kód:

```
ALTER TABLE Zamestnanec
ADD FOREIGN KEY (PSC) REFERENCES PSC(ID);
GO
```

### 3.3.4. Vloženie testovacích dát

Aby bolo možné otestovať funkcionality vytvorenej databázy je nutné vložiť nejakú vzorku testovacích dát. Dáta sa do tabuľky vkladajú pomocou príkazu INSERT INTO. Priložený kód slúži na vloženie dát do tabuľky Zamestnanec. Kompletný kód vkladania údajov je možné nájsť v prílohe č. 1.

Kód:

```
INSERT INTO Zamestnanec VALUES ('Ján', 'Malý', 'maly@gmail.com', '0904257841',
'Zahradní', '284', '11.5.1975', '1544789875', '69142');
INSERT INTO Zamestnanec VALUES ('Jana', 'Tichá', 'ticha@gmail.com', '0907584695',
'Přístavní', '24', '1.8.1987', '57896485754', '76502');
INSERT INTO Zamestnanec VALUES ('Marie', 'Smelá', 'smela.m@gmail.com',
'0902547815',
'Erbenova', '45', '11.15.1978', '5748411478', '76502');
INSERT INTO Zamestnanec VALUES ('Ondrej', 'Novák', 'novak@gmail.com',
'0907584685',
'Na Luka', '154', '5.14.1977', '51774995644', '10300');
INSERT INTO Zamestnanec VALUES ('Andrej', 'Nagy', 'nagy@gmail.com', '0907589645',
'Zkrácená', '44', '10.20.1968', '4578455655', '10300');
INSERT INTO Zamestnanec VALUES ('Anna', 'Nováková', 'novakovaAnna@gmail.com',
'0906852145',
'U Císařské cesty', '170', '7.3.1985', '7896540265', '10300');
INSERT INTO Zamestnanec VALUES ('Juraj', 'Sova', 'sova@gmail.com', '0907784524',
'Gagarinova', '46', '7.15.1987', '2458945628', '67181');
```

### 3.3.5. Pohľady

Táto kapitola obsahuje vytvorené pohľady slúžiace k zobrazeniu testovacích dát užívateľovi v prijateľnej forme. Po konzultácii so školou som vytvorila štyri pohľady, dva zamerané na študentov, jeden na zamestnancov a jeden na kurzy.

Pohľady vytvára príkaz CREATE VIEW a s vytvoreným pohľadom je možné pracovať ako s reálnymi tabuľkami databázy.

#### Pohľad na jazyky, ktoré študent ovláda

Prvý pohľad zobrazuje údaje o jazykovej zdatnosti jednotlivých študentov školy. Vo výslednom pohľade sa zobrazí meno študenta, veková kategória do ktorej spadá a jazyk s doposiaľ dosiahnutou úrovňou. Výsledok je možné vidieť na obrázku č. 17.

Kód:

```
CREATE VIEW Info_studenti AS
SELECT s.Meno + ' ' + s.Priezvisko as 'Študent', vk.Nazov as 'Veková kategória',
j.Nazov as 'Jazyk', u.Nazov
FROM Student s
LEFT JOIN Znalost_jazyka z ON z.ID_student=s.ID
LEFT JOIN Jazyk j ON z.ID_jazyk= j.ID
LEFT JOIN Uroven u ON z.ID_Uroven=u.ID
LEFT JOIN Vekova_kategoria vk ON vk.ID=s.Vekova_kat
GO
```

	Študent	Veková kategória	Jazyk	Nazov
1	Lenka Malá	19-35 rokov	Ruský jazyk	Mierne pokročilý
2	Lea Pekná	11-18 rokov	Anglický jazyk	Pokročilý
3	Laura Smelá	19-35 rokov	NULL	NULL
4	Martin Malý	6-10 rokov	Anglický jazyk	Začiatočník
5	Matej Nový	6-10 rokov	Anglický jazyk	Začiatočník
6	David Kovář	6-10 rokov	Anglický jazyk	Začiatočník

**Obrázok 17: Výpis pohľadu Info\_studenti**

(Zdroj: Vlastné spracovanie)

### Pohľad na študentov zapísaných v kurze

Účelom nasledujúceho pohľadu je zobrazenie študentov, ktorí sú zapísaní na daný kurz, v tomto prípade ide o mini kurz pre deti v pondelok. Na obrázku č. 18 vidieť výslednú tabuľku pohľadu zobrazujúcu meno študenta, kontaktné údaje (email a telefónne číslo) a stĺpec s informáciou, či študent má zaplatený kurz.

Kód:

```
CREATE VIEW Zapisany_studenti AS
SELECT s.Meno + ' ' + s.Priezvisko as 'Študent', s. Email as 'E-mail', s.Telefon
as 'Telefónne číslo',
CASE WHEN zk.Zaplatene = 1 THEN 'Áno' ELSE 'Nie' END AS 'Zaplatené'
FROM Den d JOIN Kurz k ON d.Kod=k.Den
JOIN Zapisany_kurz zk ON k.ID=zk.ID_kurz
LEFT JOIN Student s ON zk.ID_student=s.ID
WHERE k.Nazov='MiniDeti' AND d.Kod='PON'

GO
```

	Študent	E-mail	Telefónne číslo	Zaplatené
1	David Kovář	Kovar@gmail.com	0905845488	Nie
2	Martin Malý	MalyDerek@gmail.com	0903485416	Áno

**Obrázok 18: Výpis pohľadu Zapisany\_studenti**

(Zdroj: Vlastné spracovanie)

## Pohľad na zamestnancov

Tento pohľad zobrazuje údaje o zamestnancoch školy, konkrétne pohľad obsahuje meno zamestnanca, jeho kontaktné údaje (email a telefónne číslo) a pozíciu, ktorú v spoločnosti zastáva, vid' obrázok č. 19.

Kód:

```
CREATE VIEW Info_zamestnanci AS
SELECT z.Priezvisko + ' ' + z.Meno as 'Zamestnanec', z.Email, z.Telefon, p.Nazov
as 'Pozícia'
FROM Zamestnanec z
LEFT JOIN Pozicia_zamestnanca pz ON z.ID=pz.ID_zamestnanec
LEFT JOIN Pozicia p ON pz.ID_pozicia=p.ID
```

GO

	Zamestnanec	Email	Telefon	Pozícia
1	Malý Ján	maly@gmail.com	0904257841	Koordinátor
2	Tichá Jana	ticha@gmail.com	0907584695	Vedúci lektor
3	Smelá Marie	smela.m@gmail.com	0902547815	Lektor
4	Novák Ondrej	novak@gmail.com	0907584685	Riaditeľ
5	Nagy Andrej	nagy@gmail.com	0907589645	Lektor
6	Nováková Anna	novakovaAnna@gmail.com	0906852145	Lektor
7	Sova Juraj	sova@gmail.com	0907784524	Lektor

Obrázok 19: Pohľad Info\_zamestnanci

(Zdroj: Vlastné spracovanie)

## Pohľad na informácie o kurzoch

Posledný pohľad sa zameriava na základné informácie o kurzoch ako je typ kurzu, meno lektora, jazyk a úroveň, v ktorom je kurz vyučovaný, trvanie a cena kurzu. Ako v predchádzajúcich prípadoch, tak aj teraz je možné vidieť výsledok na obrázku č. 20 pod kódom pohľadu.

Kód:

```
CREATE VIEW Info_kurzy AS
SELECT DISTINCT k.Nazov as 'Názov kurzu', tk.Nazov as 'Typ kurzu', vk.Nazov as
'Veková kategória', z.Meno + ' ' + z.Priezvisko as 'Lektor', j.Nazov as 'Jazyk',
u.Nazov as 'Úroveň', tk.Trvanie, tk.Kapacita, tk.Cena as 'Cena v Kč'
FROM Typ_kurzu tk LEFT JOIN Kurz k ON tk.ID=k.Typ_kurzu
LEFT JOIN Jazyk j ON k.Jazyk=j.ID
LEFT JOIN Uroven u ON k.Uroven=u.ID
LEFT JOIN Vekova_kategoria vk ON vk.ID=k.Vekova_kat
LEFT JOIN Zapisany_kurz zk ON k.ID=zk.ID_kurz
LEFT JOIN Zamestnanec z ON zk.ID_zamestnanec=z.ID
```

GO



	Názov kurzu	Typ kurzu	Veková kategória	Lektor	Jazyk	Úroveň	Trvanie	Kapacita	Cena v Kč
1	Konverzácia Laura	Konverzácie	19-35 rokov	Anna Nováková	Ruský jazyk	Mierne pokročilý	60 minút	1	5000
2	MiniDeti	Minikurz	6-10 rokov	Jana Tichá	Anglický jazyk	Začiatočník	60 minút	5	3000
3	PETpríprava	Príprava na PET	11-18 rokov	Marie Smelá	Anglický jazyk	Pokročilý	90 minút	3	4500
4	SkDeti	Skupinový kurz	6-10 rokov	Marie Smelá	Francúzsky jazyk	Začiatočník	60 minút	14	2450
5	SkJunior	Skupinový kurz	19-35 rokov	NULL	Francúzsky jazyk	Mierne pokročilý	60 minút	14	2450
6	SkSenior	Skupinový kurz	61+ rokov	NULL	Ruský jazyk	Mierne pokročilý	60 minút	14	2450

**Obrázok 20: Pohľad Info\_kurzy**

(Zdroj: Vlastné spracovanie)

### 3.3.6. Trigger

Obsahom tejto kapitoly sú postupnosti príkazov, ktoré databáza vykoná na základe predom definovanej udalosti (DELETE, INSERT, UPDATE). Táto postupnosť sa nazýva trigger alebo spúšťač.

Pre vytvorenie spúšťača sa používa príkaz CREATE TRIGGER.

#### Trigger vymazanie študenta

Trigger je určený na vymazanie študentových údajov na základe ID študenta. Ako prvý krok je nájdenie záznamu študenta, ktorého chceme vymazať z databázy v tabuľke „deleted“. Ďalej vymaže všetky záznamy, ktoré s týmto študentom súvisia.

Kód:

```
CREATE TRIGGER vymazanie_studenta ON Student
for delete as

begin
declare @student int;
declare @jazyk int;
declare @kurz int;

select @student=ID from deleted
set @jazyk=(select top 1 ID_jazyk from Znalost_jazyka WHERE ID_student=@student);
set @kurz=(select top 1 ID_kurz from Zapisany_kurz WHERE ID_student=@student);
delete from Znalost_jazyka WHERE ID=@jazyk;
delete from Zapisany_kurz WHERE ID=@kurz;
delete from Student WHERE ID=@student;
END;

GO
```

Trigger sa spúšťa v prípade vymazania študenta, napríklad vymazaním študenta s ID 5. Nasledujúcim príkaz vykonáme vymazanie študenta:

```
DELETE FROM Student WHERE ID=5
```

### **3.4. Zhodnotenie návrhu**

Hlavným prínosom tejto práce je riešenie nedostatkov práce s evidovanými údajmi v škole. Navrhované riešenie umožňuje škole lepší prehľad o uchovávaných dátach, ktoré sú zjednotené a ako celok budú uložené na jednom mieste v databázovom systéme. Vďaka tomu sa zefektívnia administratívne procesy v škole, čo ušetrí čas a tým aj finančné prostriedky.

Návrh v aktuálnej podobe nie je vhodný pre koncových užívateľov. K systému je ešte nutné nájsť vhodné aplikačné rozhranie, ktoré bude pre užívateľov intuitívnejšie a prehľadnejšie.

Ekonomické porovnanie prínosov a nákladov riešenia v tejto podobe nie je možné vyčíslieť z dôvodu, že implementácia systému ešte neprebehla. Ale vzhľadom na to, že majiteľ si bude implementovať návrh svojpomocne, je možné počítať s tým, že celkové prínosy budú vyššie ako náklady na tento systém.

## **ZÁVER**

Hlavným cieľom tejto záverečnej práce bolo vytvoriť návrh databázového systému pre potreby jazykovej školy.

Po konzultácii s vedením školy vyšlo najavo, že škola eviduje všetky záznamy o svojich zamestnancoch, študentoch a kurzoch samostatne v Exceli bez toho, aby boli dáta medzi sebou nejako prepojené. Absenciu ucelenej evidencie údajov pomôže vyriešiť navrhnutý databázový systém, ktorý vychádza z analýzy súčasného stavu a požiadaviek školy. Systém je navrhnutý pre aktuálnej potreby školy, ale do budúca je možné ho v prípade potreby buď rozšíriť alebo prispôsobiť novým podmienkam.

Navrhnutý databázový systém uchováva všetky potrebné informácie a zároveň umožňuje efektívne a jednoducho pracovať s dátami. Nepopierateľná výhoda je v prehľadnosti dát, centralizovanom ukladaní dát a rýchlosti vyhľadávania požadovaných informácií.

## ZOZNAM POUŽITÝCH ZDROJŮV

- (1) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 3., přeprac. Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-214-3731-9.
- (2) KRÍŽ, Jiří a Petr DOSTÁL. *Databázové systémy*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.
- (3) KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
- (4) OPPEL, Andrew J. *Databáze bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.
- (5) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- (6) LAURENČÍK, Marek. *SQL: podrobný průvodce uživatele*. Praha: Grada Publishing, 2018. Průvodce (Grada). ISBN 978-80-271-0774-2.
- (7) LACKO, Luboslav. *Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]*. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.
- (8) POKORNÝ, Jaroslav a Michal VALENTA. *Databázové systémy*. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.
- (9) OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.
- (10) GROFF, James R. a Paul N. WEINBERG. *SQL: kompletní průvodce*. Brno: CP Books, 2005. Programování (CP Books). ISBN 80-251-0369-2.
- (11) LAURENČÍK, Marek. *Tvorba www stránek v HTML a CSS*. Praha: Grada Publishing, 2019. Průvodce (Grada). ISBN 978-80-271-2241-7.

## ZOZNAM POUŽITÝCH SKRATIEK A SYMBOLOV

CAE	Cambridge Advanced Certificate in English
DCL	Data Control Language
DDL	Data Definition Language
DML	Data Manipulation Language
DQL	Data Query Language
FCE	First Certificate in English
FK	Foreign Key (cudzí kľúč)
MS	Microsoft
Napr.	Napríklad
PET	Preliminary English Test
PK	Primary Key (primárny kľúč)
PSČ	Poštové Smerovacie Číslo
SQL	Structured Query Language
Tzv.	Takzvaný

## ZOZNAM OBRÁZKOV

Obrázok 1: Znázornenie informácie .....	12
Obrázok 2: Znázornenie dát.....	13
Obrázok 3: Zložky databázového systému .....	14
Obrázok 4: Lineárny dátový model .....	15
Obrázok 5: Hierarchický dátový model.....	16
Obrázok 6: Sieťový dátový model.....	17
Obrázok 7: Relačný dátový model.....	17
Obrázok 8: Objektový dátový model.....	18
Obrázok 9: Vzťah 1:1 .....	20
Obrázok 10: Vzťah 1:N a N:1.....	21
Obrázok 11: Vzťah N:M.....	21
Obrázok 12: Dekompozícia vzťahu N:M .....	22
Obrázok 13: Organizačná štruktúra jazykovej školy .....	31
Obrázok 14: Vývojový diagram prihlásenia na kurz .....	33
Obrázok 15: Vývojový diagram evidencie študentov.....	34
Obrázok 16: ER-diagram .....	49
Obrázok 17: Výpis pohľadu Info_studenti .....	53
Obrázok 18: Výpis pohľadu Zapisany_studenti .....	53
Obrázok 19: Pohľad Info_zamestnanci.....	54
Obrázok 20: Pohľad Info_kurzy .....	55

## ZOZNAM TABULIEK

Tabuľka 1: Tabuľka porušujúca 1. normálnu formu .....	23
Tabuľka 2: 1. normálna forma .....	23
Tabuľka 3: Tabuľka porušujúca 2. normálnu formu .....	24
Tabuľka 4: 1. tabuľka dekompozície .....	24
Tabuľka 5: 2. tabuľka dekompozície .....	24
Tabuľka 6: Tabuľka porušujúca 3. normálnu formu .....	25
Tabuľka 7: 1. tabuľka dekompozície .....	25
Tabuľka 8: 2. tabuľka dekompozície .....	26
Tabuľka 9: Identifikácia entít .....	35
Tabuľka 10: Identifikácia vzťahov .....	36
Tabuľka 11: Atribúty tabuľky Zamestnanec.....	37
Tabuľka 12: Atribúty tabuľky Študent .....	38
Tabuľka 13: Atribúty tabuľky Kurz.....	39
Tabuľka 14: Atribúty tabuľky Typ kurzu .....	40
Tabuľka 15: Atribúty tabuľky Jazyk.....	41
Tabuľka 16: Atribúty tabuľky Úroveň.....	41
Tabuľka 17: Atribúty tabuľky Miestnosť .....	42
Tabuľka 18: Atribúty tabuľky Pozícia.....	43
Tabuľka 19: Atribúty tabuľky Deň .....	43
Tabuľka 20: Atribúty tabuľky Veková kategória .....	44
Tabuľka 21: Atribúty tabuľky PSČ.....	44
Tabuľka 22: Atribúty tabuľky Zapísaný kurz.....	45
Tabuľka 23: Atribúty tabuľky Vyučovaný jazyk.....	46
Tabuľka 24: Atribúty tabuľky Pozícia zamestnanca .....	46
Tabuľka 25: Atribúty tabuľky Znalosť jazyka.....	47

## **ZOZNAM PRÍLOH**

Príloha č. 1: Zdrojový SQL kód databázy .....	I
---	---



## Príloha č. 1: Zdrojový SQL kód databázy

-- Vytvorenie databázy

```
CREATE DATABASE Bakalarska_praca
```

-- Vytvorenie tabuliek

```
CREATE TABLE Den
```

```
(  
Kod varchar(3) NOT NULL ,  
Nazov varchar(10) NOT NULL ,  
CONSTRAINT PK_Den PRIMARY KEY (kod)  
);  
GO
```

```
CREATE TABLE Jazyk
```

```
(  
ID int NOT NULL identity(1,1),  
Nazov varchar(20) NOT NULL ,  
CONSTRAINT PK_Jazyk PRIMARY KEY (ID)  
);  
GO
```

```
CREATE TABLE Kurz
```

```
(  
ID int NOT NULL identity(1,1),  
Nazov varchar(50) NOT NULL ,  
Cas time(0) NOT NULL ,  
Miestnost int NOT NULL ,  
Den varchar(3) NOT NULL ,
```

```
Typ_kurzu int NOT NULL ,
Jazyk int NOT NULL ,
Uroven int NOT NULL ,
Vekova_kat int NOT NULL ,
CONSTRAINT PK_Kurz PRIMARY KEY (ID),
);
GO
```

```
CREATE TABLE Miestnost
(
ID int NOT NULL identity(1,1),
Nazov varchar(20) NOT NULL ,
Popis varchar(100),
Kapacita int NOT NULL ,
CONSTRAINT PK_Miestnost PRIMARY KEY (ID)
);
GO
```

```
CREATE TABLE Pozicia
(
ID int NOT NULL identity(1,1),
Nazov varchar(30) NOT NULL ,
Popis varchar(200),
CONSTRAINT PK_Pozicia PRIMARY KEY (ID)
);
GO
```

```
CREATE TABLE Pozicia_zamestnanca
(
ID int NOT NULL identity(1,1),
ID_zamestnanec int NOT NULL ,
```

```
ID_pozicia int NOT NULL ,  
CONSTRAINT PK_Pozicia_zamestnanca PRIMARY KEY (ID)  
);  
GO
```

```
CREATE TABLE PSC  
(  
ID varchar(6) NOT NULL ,  
Mesto varchar(25) NOT NULL ,  
CONSTRAINT PK_PSC PRIMARY KEY (ID)  
);  
GO
```

```
CREATE TABLE Typ_kurzu  
(  
ID int NOT NULL identity(1,1),  
Nazov varchar(50) NOT NULL ,  
Trvanie varchar(10) NOT NULL ,  
Cena int NOT NULL ,  
Kapacita int NOT NULL ,  
CONSTRAINT PK_Typ_kurzu PRIMARY KEY (ID)  
);  
GO
```

```
CREATE TABLE Vekova_kategoria  
(  
ID int NOT NULL identity(1,1),  
Nazov varchar(20) NOT NULL ,  
CONSTRAINT PK_Vekova_kategoria PRIMARY KEY (ID)  
);  
GO
```

```
CREATE TABLE Vyucoваны_jazyk
(
  ID int NOT NULL identity(1,1),
  poznamky varchar(200) ,
  ID_zamestnanec int NOT NULL ,
  ID_uroven int NOT NULL ,
  ID_jazyk int NOT NULL ,
  CONSTRAINT PK_Vyucoваны_jazyk PRIMARY KEY (ID)
);
GO
```

```
CREATE TABLE Zamestnanec
(
  ID int NOT NULL identity(1,1),
  Meno varchar(20) NOT NULL ,
  Priezvisko varchar(30) NOT NULL ,
  Email varchar(50) ,
  Telefon varchar(12) NOT NULL ,
  Ulica varchar(20) NOT NULL ,
  Popisne_c varchar(10) NOT NULL ,
  Datu_nar date NOT NULL ,
  Cislo_uctu varchar(20) NOT NULL ,
  PSC varchar(6) NOT NULL ,
  CONSTRAINT PK_Zamestnanec PRIMARY KEY (ID),
);
GO
```

```
CREATE TABLE Zapisany_kurz
(
  ID int NOT NULL identity(1,1),
```

```
poznamky varchar(200),
ID_student int NOT NULL ,
ID_zamestnanec int NOT NULL ,
ID_kurz int NOT NULL ,
Zaplatene BIT NOT NULL DEFAULT 0 ,
CONSTRAINT PK_Zapisany_kurz PRIMARY KEY (ID),
);
GO
```

```
CREATE TABLE Znalost_jazyka
(
ID int NOT NULL identity(1,1),
poznamky varchar(200) ,
ID_student int NOT NULL ,
ID_jazyk int NOT NULL ,
ID_uroven int NOT NULL ,
CONSTRAINT PK_Znalost_jazyka PRIMARY KEY (ID),
);
GO
```

```
CREATE TABLE Uroven
(
ID int NOT NULL identity(1,1),
Nazov varchar(20) NOT NULL ,
Popis varchar(200) ,
CONSTRAINT PK_Uroven PRIMARY KEY (ID)
);
GO
```

```
CREATE TABLE Student
(
```

```
ID int NOT NULL identity(1,1) ,
Meno varchar(20) NOT NULL ,
Priezvisko varchar(30) NOT NULL ,
Email varchar(50) NOT NULL ,
Telefon varchar(12) ,
Vekova_kat int NOT NULL ,
CONSTRAINT PK_Student PRIMARY KEY (ID),
);
GO
```

-- Vytvorenie cudzích kľúčov

```
ALTER TABLE Kurz ADD
FOREIGN KEY (Miestnost) REFERENCES Miestnost(ID),
FOREIGN KEY (Den) REFERENCES Den(kod),
FOREIGN KEY (Typ_kurzu) REFERENCES Typ_kurzu(ID),
FOREIGN KEY (Jazyk) REFERENCES Jazyk(ID),
FOREIGN KEY (Uroven) REFERENCES Uroven(ID),
FOREIGN KEY (Vekova_kat) REFERENCES Vekova_kategoria(ID);
GO
```

```
ALTER TABLE Pozicia_zamestnanca ADD
FOREIGN KEY (ID_zamestnanec) REFERENCES Zamestnanec(ID),
FOREIGN KEY (ID_pozicia) REFERENCES Pozicia(ID);
GO
```

```
ALTER TABLE Vyučovany_jazyk ADD
FOREIGN KEY (ID_zamestnanec) REFERENCES Zamestnanec(ID),
FOREIGN KEY (ID_uroven) REFERENCES Uroven(ID),
FOREIGN KEY (ID_jazyk) REFERENCES Jazyk(ID);
GO
```

```
ALTER TABLE Zamestnanec
ADD FOREIGN KEY (PSC) REFERENCES PSC(ID);
GO
```

```
ALTER TABLE Zapisany_kurz ADD
FOREIGN KEY (ID_student) REFERENCES Student(ID),
FOREIGN KEY (ID_zamestnanec) REFERENCES Zamestnanec(ID),
FOREIGN KEY (ID_kurz) REFERENCES Kurz(ID);
GO
```

```
ALTER TABLE Znalost_jazyka ADD
FOREIGN KEY (ID_student) REFERENCES Student(ID),
FOREIGN KEY (ID_jazyk) REFERENCES Jazyk(ID),
FOREIGN KEY (ID_uroven) REFERENCES Uroven(ID);
GO
```

```
ALTER TABLE Student
ADD FOREIGN KEY (Vekova_kat) REFERENCES Vekova_kategoria(ID);
GO
```

```
-- Vkladanie dát
```

```
INSERT INTO Jazyk VALUES ('Anglický jazyk');
INSERT INTO Jazyk VALUES ('Francúzsky jazyk');
INSERT INTO Jazyk VALUES ('Ruský jazyk');
```

```
INSERT INTO Uroven VALUES ('Začiatocník', '');
INSERT INTO Uroven VALUES ('Mierne pokročilý', 'Študent ovláda základnú
gramatiku, slovná zásoba obsahuje 750-1000 slov');
INSERT INTO Uroven VALUES ('Pokročilý', '');
```

INSERT INTO Uroven VALUES ('Veľmi pokročilý', 'Študent chápe odborným textom a dokáže sa plynule vyjadrovať ku všetkým témam, ktorým rozumie');

INSERT INTO Den VALUES ('PON', 'Pondelok');

INSERT INTO Den VALUES ('UTO', 'Utorok');

INSERT INTO Den VALUES ('STR', 'Streda');

INSERT INTO Den VALUES ('STV', 'Štvrtok');

INSERT INTO Den VALUES ('PIA', 'Piatok');

INSERT INTO Typ\_kurzu VALUES ('Minikurz', '60 minút', '3000', '5');

INSERT INTO Typ\_kurzu VALUES ('Príprava na PET', '90 minút', '4500', '3');

INSERT INTO Typ\_kurzu VALUES ('Skupinový kurz', '60 minút', '2450', '14');

INSERT INTO Typ\_kurzu VALUES ('Konverzácie', '60 minút', '5000', '1');

INSERT INTO Miestnost VALUES ('U115', 'Učebňa 1', '10');

INSERT INTO Miestnost VALUES ('S210', 'Sklad', '2');

INSERT INTO Miestnost VALUES ('U120', 'PC učebňa', '5');

INSERT INTO Miestnost VALUES ('U220', 'Učebňa 2', '20');

INSERT INTO Vekova\_kategoria VALUES ('6-10 rokov');

INSERT INTO Vekova\_kategoria VALUES ('11-18 rokov');

INSERT INTO Vekova\_kategoria VALUES ('19-35 rokov');

INSERT INTO Vekova\_kategoria VALUES ('36-60 rokov');

INSERT INTO Vekova\_kategoria VALUES ('61+ rokov');

INSERT INTO PSC VALUES ('69142', 'Valtice');

INSERT INTO PSC VALUES ('10300', 'Praha - Benice');

INSERT INTO PSC VALUES ('76502', 'Otrokovice');

INSERT INTO PSC VALUES ('67181', 'Znojmo');



```

INSERT INTO Pozicia VALUES ('Riaditel', '');
INSERT INTO Pozicia VALUES ('Vedúci lektor', 'Zodpovednosť za lektorov');
INSERT INTO Pozicia VALUES ('Kordinátor', 'Komunikácia so študentmi a
lektormi');
INSERT INTO Pozicia VALUES ('Lektor', 'Vyučovanie jazykov');

INSERT INTO Kurz VALUES ('MiniDeti', '16:30', '1', 'PON', '1', '1', '1', '1');
INSERT INTO Kurz VALUES ('SkJunior', '16:00', '4', 'STR', '3', '2', '2', '3');
INSERT INTO Kurz VALUES ('PETpríprava', '17:30', '3', 'STV', '2', '1', '3', '2');
INSERT INTO Kurz VALUES ('SkDeti', '17:00', '4', 'STV', '3', '2', '1', '1');
INSERT INTO Kurz VALUES ('SkSenior', '14:00', '4', 'PON', '3', '3', '2', '5');
INSERT INTO Kurz VALUES ('Konverzácia Laura', '17:00', '1', 'PIA', '4', '3', '2', '3');

INSERT INTO Zamestnanec VALUES ('Ján', 'Malý', 'maly@gmail.com', '0904257841',
'Zahradní', '284', '11.5.1975', '1544789875', '69142');
INSERT INTO Zamestnanec VALUES ('Jana', 'Tichá', 'ticha@gmail.com',
'0907584695', 'Přistavní', '24', '1.8.1987', '57896485754', '76502');
INSERT INTO Zamestnanec VALUES ('Marie', 'Smelá', 'smela.m@gmail.com',
'0902547815', 'Erbenova', '45', '11.15.1978', '5748411478', '76502');
INSERT INTO Zamestnanec VALUES ('Ondrej', 'Novák', 'novak@gmail.com',
'0907584685', 'Na Luka', '154', '5.14.1977', '51774995644', '10300');
INSERT INTO Zamestnanec VALUES ('Andrej', 'Nagy', 'nagy@gmail.com',
'0907589645', 'Zkráčená', '44', '10.20.1968', '4578455655', '10300');
INSERT INTO Zamestnanec VALUES ('Anna', 'Nováková',
'novakovaAnna@gmail.com', '0906852145', 'U Císařské cesty', '170', '7.3.1985',
'7896540265', '10300');
INSERT INTO Zamestnanec VALUES ('Juraj', 'Sova', 'sova@gmail.com',
'0907784524', 'Gagarinova', '46', '7.15.1987', '2458945628', '67181');

INSERT INTO Pozicia_zamestnanca VALUES ('4', '1');
INSERT INTO Pozicia_zamestnanca VALUES ('2', '2');
INSERT INTO Pozicia_zamestnanca VALUES ('3', '4');
INSERT INTO Pozicia_zamestnanca VALUES ('1', '3');
INSERT INTO Pozicia_zamestnanca VALUES ('5', '4');

```

INSERT INTO Pozicia\_zamestnanca VALUES ('6', '4');

INSERT INTO Pozicia\_zamestnanca VALUES ('7', '4');

INSERT INTO Student VALUES ('Lenka', 'Malá', 'mala@gmail.com', '0905841258', '3');

INSERT INTO Student VALUES ('Lea', 'Pekná', 'lea.p@gmail.com', '0905687422', '2');

INSERT INTO Student VALUES ('Laura', 'Smelá', 'smelaLaura@gmail.com', '0908475621', '3');

INSERT INTO Student VALUES ('Martin', 'Malý', 'MalyDerek@gmail.com', '0903485416', '1');

INSERT INTO Student VALUES ('Matej', 'Nový', 'Nová.Katarína@gmail.com', '0905984525', '1');

INSERT INTO Student VALUES ('David', 'Kovář', 'Kovar@gmail.com', '0905845488', '1');

INSERT INTO Vyucovany\_jazyk VALUES ('', '2', '4', '1');

INSERT INTO Vyucovany\_jazyk VALUES ('', '2', '3', '3');

INSERT INTO Vyucovany\_jazyk VALUES ('', '3', '4', '1');

INSERT INTO Vyucovany\_jazyk VALUES ('', '3', '4', '2');

INSERT INTO Vyucovany\_jazyk VALUES ('', '4', '4', '3');

INSERT INTO Znalost\_jazyka VALUES ('', '5', '1', '1');

INSERT INTO Znalost\_jazyka VALUES ('', '4', '1', '1');

INSERT INTO Znalost\_jazyka VALUES ('', '6', '1', '1');

INSERT INTO Znalost\_jazyka VALUES ('', '2', '1', '3');

INSERT INTO Znalost\_jazyka VALUES ('', '1', '3', '2');

INSERT INTO Zapisany\_kurz VALUES ('', '2', '3', '3', '1');

INSERT INTO Zapisany\_kurz VALUES ('', '6', '2', '1', '');

INSERT INTO Zapisany\_kurz VALUES ('', '4', '2', '1', '1');

INSERT INTO Zapisany\_kurz VALUES ('', '5', '3', '4', '');

INSERT INTO Zapisany\_kurz VALUES ('', '6', '3', '4', '');

INSERT INTO Zapisany\_kurz VALUES ('', '3', '6', '6', '1');

-- Pohľady

-- Výpis jazykov, ktoré študent ovláda a na akej úrovni

```
CREATE VIEW Info_studenti AS
```

```
SELECT s.Meno + ' ' + s.Priezvisko as 'Študent', vk.Nazov as 'Veková kategória',  
j.Nazov as 'Jazyk', u.Nazov
```

```
FROM Student s
```

```
LEFT JOIN Znalost_jazyka z ON z.ID_student=s.ID
```

```
LEFT JOIN Jazyk j ON z.ID_jazyk= j.ID
```

```
LEFT JOIN Uroven u ON z.ID_Uroven=u.ID
```

```
LEFT JOIN Vekova_kategoria vk ON vk.ID=s.Vekova_kat
```

```
GO
```

-- Výpis študentov zapísaných na minikurz pre deti s ich kontaktnými údajmi

```
CREATE VIEW Zapisany_studenti AS
```

```
SELECT s.Meno + ' ' + s.Priezvisko as 'Študent', s. Email as 'E-mail', s.Telefon as  
'Telefónne číslo',
```

```
CASE WHEN zk.Zaplatene = 1 THEN 'Áno' ELSE 'Nie' END AS 'Zaplatené'
```

```
FROM Den d JOIN Kurz k ON d.Kod=k.Den
```

```
JOIN Zapisany_kurz zk ON k.ID=zk.ID_kurz
```

```
LEFT JOIN Student s ON zk.ID_student=s.ID
```

```
WHERE k.Nazov='MiniDeti' AND d.Kod='PON'
```

```
GO
```

-- Výpis všetkých zamestnancov školy, ich kontaktnými údajmi a pozíciou ktorú zastávajú

```
CREATE VIEW Info_zamestnanci AS
```

```

SELECT z.Priezvisko + ' ' + z.Meno as 'Zamestnanec', z.Email, z.Telefon, p.Nazov as
'Pozícia'
FROM Zamestnanec z
LEFT JOIN Pozicia_zamestnanca pz ON z.ID=pz.ID_zamestnanec
LEFT JOIN Pozicia p ON pz.ID_pozicia=p.ID
GO

```

-- Výpis všetkých kurzov

```

CREATE VIEW Info_kurzy AS
SELECT DISTINCT k.Nazov as 'Názov kurzu', tk.Nazov as 'Typ kurzu', vk.Nazov as
'Veková kategória',
z.Meno + ' ' + z.Priezvisko as 'Lektor',j.Nazov as 'Jazyk', u.Nazov as 'Úroveň',
tk.Trvanie,
tk.Kapacita,tk.Cena as 'Cena v Kč'
FROM Typ_kurzu tk LEFT JOIN Kurz k ON tk.ID=k.Typ_kurzu
LEFT JOIN Jazyk j ON k.Jazyk=j.ID
LEFT JOIN Uroven u ON k.Uroven=u.ID
LEFT JOIN Vekova_kategoria vk ON vk.ID=k.Vekova_kat
LEFT JOIN Zapisany_kurz zk ON k.ID=zk.ID_kurz
LEFT JOIN Zamestnanec z ON zk.ID_zamestnanec=z.ID
GO

```

--Trigger vymazanie študenta

```

CREATE TRIGGER vymazanie_studenta ON Student
for delete as

begin
declare @student int;
declare @jazyk int;
declare @kurz int;

```

```
select @student=ID from deleted
set @jazyk=(select top 1 ID_jazyk from Znalost_jazyka WHERE
ID_student=@student);
set @kurz=(select top 1 ID_kurz from Zapisany_kurz WHERE ID_student=@student);
delete from Znalost_jazyka WHERE ID=@jazyk;
delete from Zapisany_kurz WHERE ID=@kurz;
delete from Student WHERE ID=@student;
END;
GO
DELETE FROM Student WHERE ID=5
```