



Bakalářská práce

Řešení problematiky informačního systému střední školy

Studijní program:

Studijní obor:

Autor práce:

Vedoucí práce:

B0613A140005 – Informační technologie

Aplikovaná informatika

Daniel Adámek

Ing. Lenka Kosková Třísková Ph.D.

Liberec 2024

Tento list nahradte
originálem zadání.

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

17. 6. 2024

Daniel Adámek

Řešení problematiky informačního systému střední školy

Abstrakt

Tato bakalářská práce se primárně zaměřuje na návrh a implementaci nových komponent pro řešení stávajícího informačního systému střední průmyslové školy, s cílem eliminovat identifikované slabiny a využít potenciál pro zlepšení. Práce vychází z podrobné analýzy současného stavu systému, včetně jeho funkcionalit, toku dat, synchronizačních procesů, API rozhraní, legislativních požadavků a bezpečnostních opatření. Na základě zjištěných výsledků se práce soustředí na návrh a implementaci klíčových funkcí pro efektivnější správu školy a odbourání automatizovatelných a repetitivních úkolů. Důraz je kladen na praktickou aplikovatelnost navrhovaných řešení, jejich integraci do stávajícího systému a možnost dalšího rozvoje.

Klíčová slova: informační systém, návrh, in-house vývoj, školství

Abstract

This bachelor thesis primarily focuses on the design and implementation of new components for the solution of the existing information system of a secondary industrial school, with the aim of eliminating identified weaknesses and exploiting the potential for improvement. The work is based on a detailed analysis of the current state of the system, including its functionalities, data flow, synchronization processes, API interfaces, legislative requirements and security measures. Based on the findings, the thesis focuses on the design and implementation of key features for more efficient school management and the elimination of automated and repetitive tasks. Emphasis is placed on the practical applicability of the proposed solutions, their integration into the existing system and the possibility of further development.

Keywords: information system, design, in-house development, education

Poděkování

Prvně bych rád vyjádřil svou vděčnost Střední průmyslové škole elektrotechnické, Praha 2, Ječná 30, kterou zastupuje ředitel Ing. Bc. et Bc. Ondřej Mandík ING-PAED IGIP. Bez jeho laskavosti a podpory by toto dílo nebylo možné realizovat. Škola mi poskytla nezbytné informace, umožnila mi upravit svůj informační systém.. Tato zkušenost byla pro mě nesmírně cenná a pomohla mi v mnoha studijních i pedagogických aspektech.

Zvláštní poděkování patří všem zaměstnancům školy a mým kolegům pedagogům, kteří mi pomohli kritickým pohledem na stávající systém a při hledání nových, lepších řešení. Jejich nápady a zpětná vazba byly jedním z klíčových aspektů pro vytvoření nového informačního systému, který je efektivní.

Dále bych chtěl poděkovat všem členům mého vedení, rodině a přátelům, kteří mi poskytli cennou pomoc a podporu během procesu vytváření této bakalářské práce. Jejich trpělivost, porozumění a povzbuzování bylo pro mě během celého procesu klíčové.

Nakonec bych chtěl vyjádřit svou vděčnost všem, kteří se na tomto díle podíleli, ať už přímo nebo nepřímo. Bez jejich kolektivního úsilí a podpory by tento projekt nebyl možný. Vaše práce a podpora byla cenná a jsem vám za to hluboce vděčný.

Děkuji všem.

Obsah

Obsah	6
Seznam zkratek	7
1 Agenda tříd	8
1.1 Popis problému	8
1.2 Úvod do problematiky	8
1.3 Požadavky	8
1.4 Návrhy implementace	9
2 Agenda žákovských skupin	16
2.1 Definice	16
2.2 Podmínky pro skupiny	16
2.3 Kategorizace podmnožin	17
2.4 Důsledky a diskuse	18
2.5 Příklady	19
2.6 Implementace	21
3 tridnice	26
4 Reference	27
5 Přílohy	28

Seznam zkratek

FM TUL	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
SPŠE Ječná	Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30
MŠMT	Ministerstvo školství, mládeže a tělovýchovy
IS	Informační systém
ŠIS	Školní informační systém
OVM	Orgán veřejné moci
NSESSS	Národní standard pro elektronické systémy spisové služby
UCD	User Centered Design

1 Agenda tříd

1.1 Popis problému

Implementace agendy tříd v rozvrhovém API školního systému představuje komplexní úkol, jehož řešení vyžaduje pečlivé zvážení a hluboké pochopení jak administrativních aspektů pro správu školy, tak technických. Tento problém je esenciální pro správné fungování školního systému, který musí být schopen efektivně spravovat nejen aktuální rozvrhy a třídy, ale také uchovávat historická data.

1.2 Úvod do problematiky

Školní rozvrhový systém je srdcem administrativy školy, zajišťující, že učitelé, žáci a učebny jsou správně přiděleni v rámci různých časových bloků během školního roku. Tento systém musí zvládat nejen plánování aktuálních rozvrhů, ale také uchovávání historických dat.

1.3 Požadavky

Pro lepší pochopení problému je důležité nahlédnout do širších souvislostí. Školní systém musí splňovat následující požadavky:

- **Konzistence dat:** Zajištění, že data o třídách, rozvrzích a žácích jsou konzistentní napříč různými časovými obdobími.
- **Historická data:** Uchovávání historických záznamů pro administrativní účely.
- **Flexibilita:** Systém musí být schopen flexibilně reagovat na změny, jako je přechod žáků mezi třídami, změny učitelů a učeben.

V kontextu školního rozvrhového systému se agenda tříd týká správy informací o třídách, které zahrnují aktuální třídy, jejich žáky, rozvrhy, záznamy probrané látky (?? třídnice) a absence. Systém musí být schopen všechny změny správně zaznamenat a reflektovat. Zmíněný úkol je komplikován několika faktory:

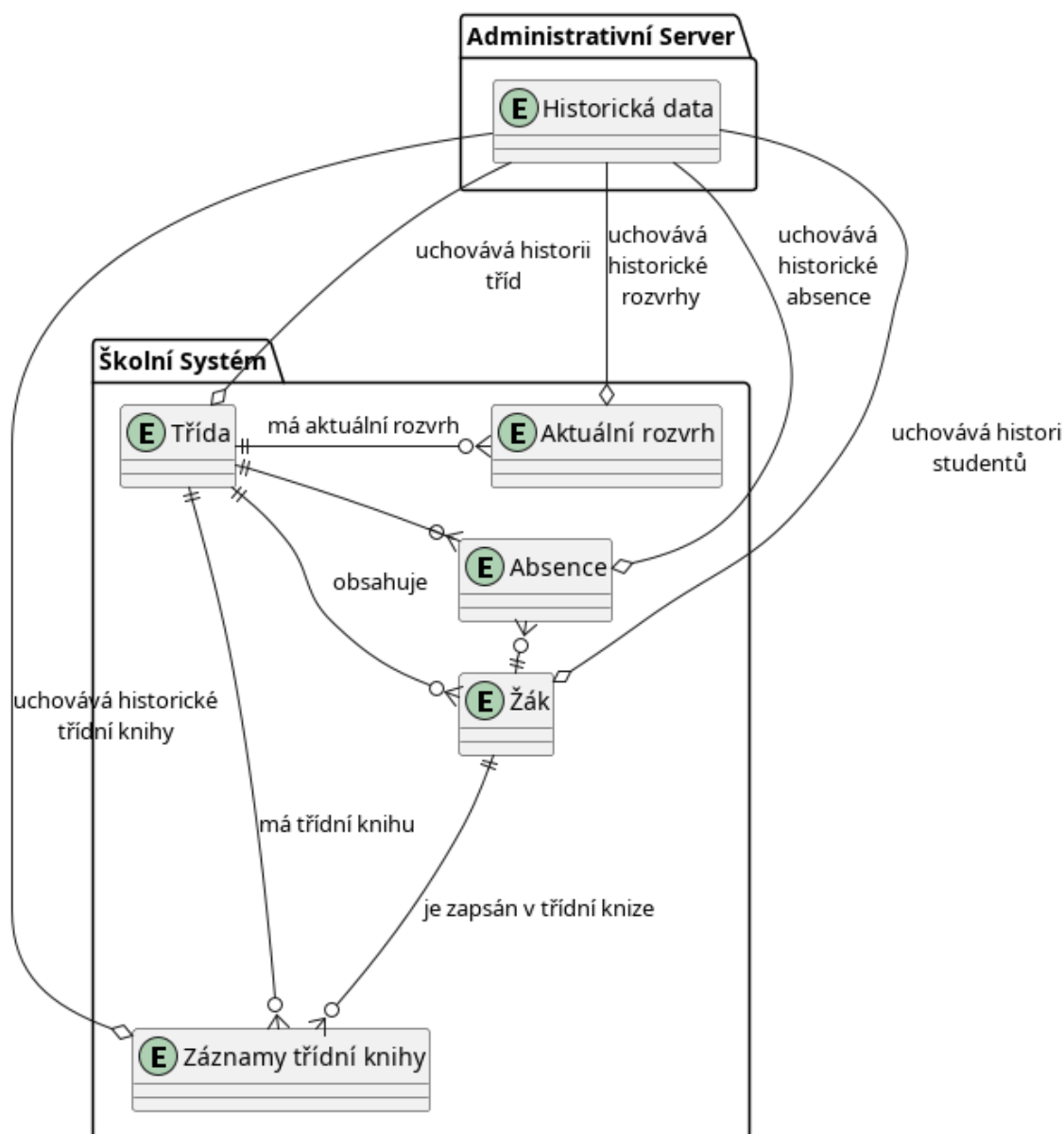
1. **Horizontální přechody studentů:** Žák během studia v průběhu školního roku může změnit třídu, obor nebo třeba školu za specifických podmínek, který stanovuje školský zákon a ředitel školy.
2. **Vertikální přechody studentů:** Žák, který nesplní zákonné požadavky pro postup do dalšího ročníku, tedy neprospěl, může být zařazen v následujícím školním roce do jiné třídy a ročník opakovat. Žák též může vzdělávání přerušit, či školu opustit.
3. **Přerušení studia:** Student může během probíhajícího studia požádat ředitele školy o přerušení vzdělávání. Maximální doba trvání přerušení dle zákona § 66 odst. 5 zákona č. 561/2004 Sb. školský zákon - znění od 01.01.2024 [1] až na dobu dvou let s výjimkou mateřství [2], které se řídí jiným procesem.
4. **Administrativní a kontrolní požadavky:** Uchovávání dat pro administrativní, právní a legislativní důvody, kterým mohou být inspekce státními orgány nebo například soudy.

Z těchto důvodů je implementace agendy tříd ve školním rozvrhovém systému složitá a vyžaduje pečlivé zvážení různých přístupů. Je důležité navrhnout systém, který bude nejen efektivní a flexibilní, ale také bude splňovat všechny uvedené požadavky. V následujících částech této kapitoly se budeme podrobněji zabývat jednotlivými návrhy implementace a jejich dopady na celkovou konzistenci a funkčnost školního systému.

1.4 Návrhy implementace

Na základě výše uvedených požadavků a kontextu jsou navrženy tři hlavní přístupy k implementaci agendy tříd v rozvrhovém API:

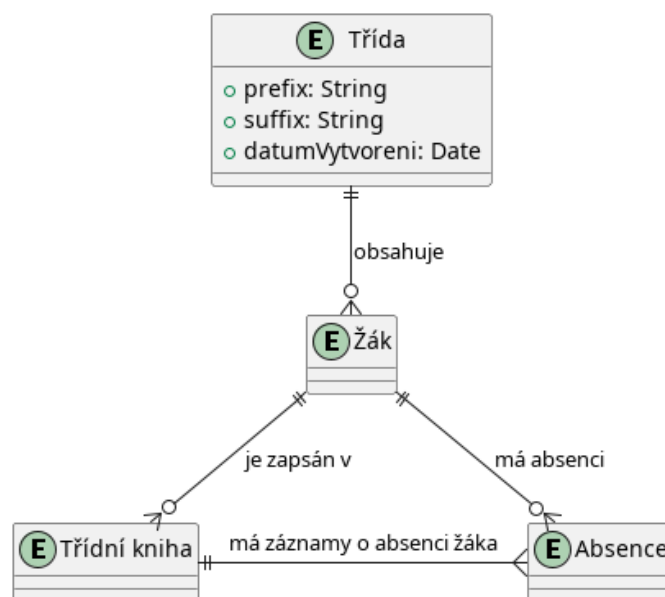
1. **Ukládání pouze aktuálního ročníku a aktuálních rozvrhů:** Tento přístup minimalizuje datovou zátěž na serveru tím, že uchovává pouze informace o aktuálním ročníku a rozvrzích. Historická data jsou uložena na odděleném administrativním serveru.



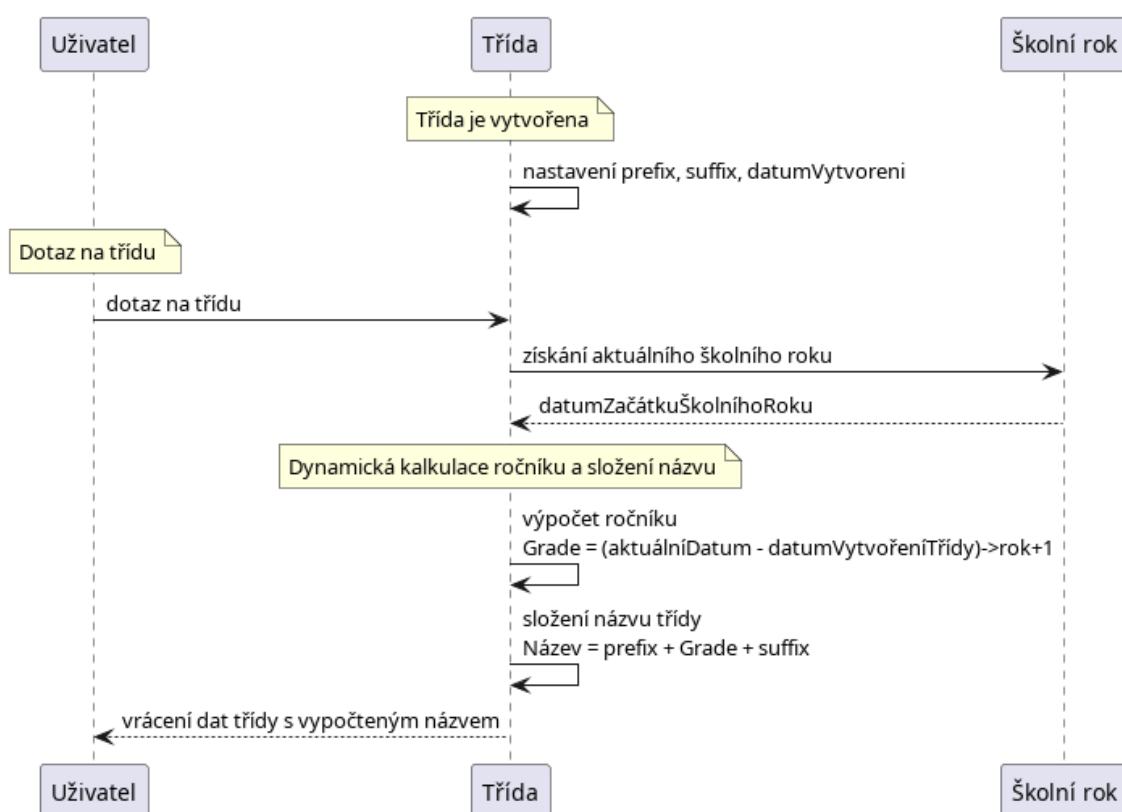
Obrázek 1.1: Ukládání pouze aktuálního ročníku a aktuálních rozvrhů

Před začátkem každého školního roku by bylo tedy potřeba sjednat synchronizační proces pro archivování dat na administrativní server, smazání z rozvrhového serveru a nahrání nových dat. Tento synchronizační mechanismus by mohl být obtížný pro správný chod. Předpokládáme, že oba servery spravuje jiný zaměstnanec. V případě chyby v programu způsobené programátory by bylo možné nenávratně přijít o důležitá data.

2. **Třída definována prefixem, suffixem, datem vytvoření a kalkulovaným ročníkem:** Tento přístup ukládá informace o třídách pomocí kombinace prefixu, suffixu a data vytvoření třídy. Ročník je kalkulován jako rozdíl mezi datem vytvoření a začátkem aktuálního školního roku.



Obrázek 1.2: Relační schéma varianty s kalkulovaným ročníkem



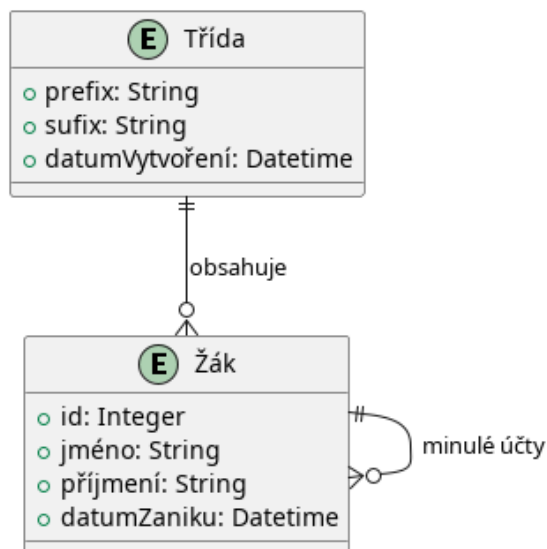
Obrázek 1.3: Schéma průběhu dotazu na jméno třídy

Problémem v tomto případě je samotná kalkulace. Pro dodržení aktuálnosti dat, musí být ročník vždy vypočítán, protože jakákoliv forma uchovávání

mezivýsledku by mohla vést k nekonzistenci v čase. Pokud dorazí požadavek k získání aktuálních tříd, musí server vypočítat všechny ročníky a spojit celé jméno třídy.

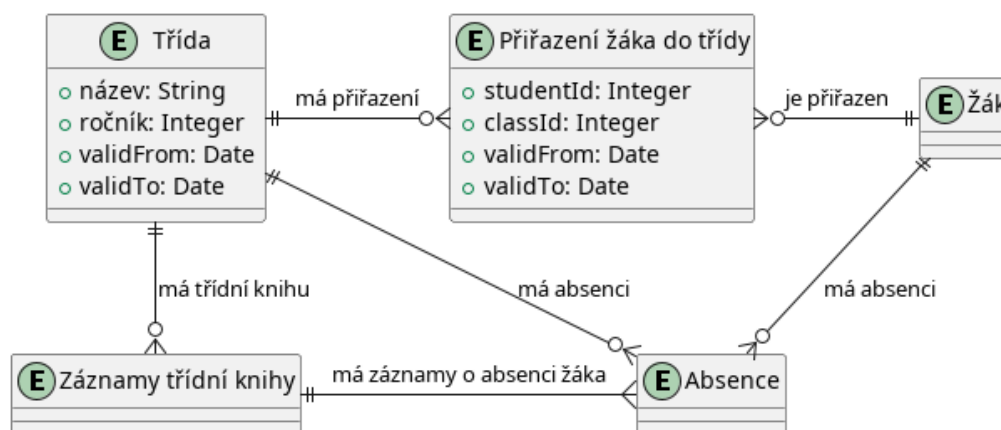
Dalším problémem tohoto řešení jsou propadlí nebo přecházející žáci. Jakmile žák propadne, nastávají 2 varianty řešení:

- odebrat žáka z aktuální třídy a přiřadit jej do nové třídy, ale ztratí minulé záznamy, protože třída je jedním z atributů žáka.
Řešením může být např. psané záznamy o změnách, avšak nepokryjí všechny požadavky pro snadnou manipulaci s daty.
- vytvoření nového žáka, který začíná studovat v pokročilém ročníku. Tím by byla zajištěna konzistence dat, avšak duplicitní účty pro jednoho žáka.
Řešením může být propojení minulých účtů žáka s novými (relací 1:N), kde žák by měl uložené své minulé účty. Stále je problém, že to není řešení příčiny, ale důsledku.

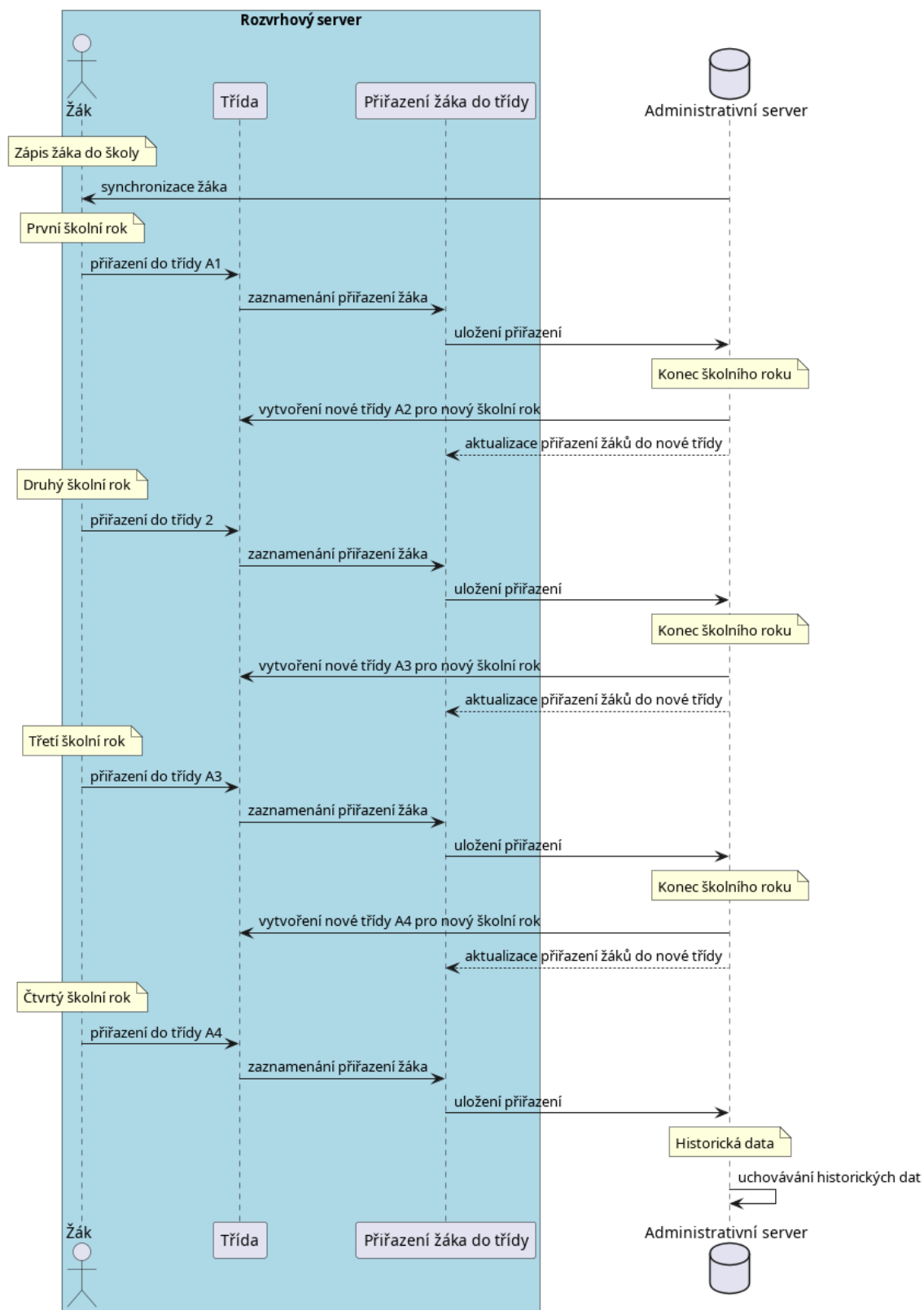


Obrázek 1.4: Schéma vazby na předchozí účty

3. **Třída definována celým názvem a datem validity od - do:** Tento přístup zahrnuje kompletní informace o třídě včetně ročníku a platnosti od-do. Každý rok se vytváří nové třídy a vztahy mezi studenty a třídami jsou zaznamenány pomocí M:N tabulky.



Obrázek 1.5: Schéma vazby M:N pro přiřazení třídy v návrhu od - do



Obrázek 1.6: Schéma průchodu žáka školním procesem z podlehu rozvrhů v návrhu validity od - do

Hlavní výhodou tohoto řešení je konzistence dat i při přesunu žáků v třídách. V tom případě stačí ukončit validitu dotčených tříd, vytvořit nové se stejným názvem a přiřazení nových žáků ve správném pořadí.

Nevýhodou je vyšší náročnost dotazování na třídu konkrétního žáka. Databáze musí projít všechny přiřazené třídy žáka, vybrat tu aktuálně validní dle dat validity a následně třídu vybrat.

2 Agenda žákovských skupin

Ve vzdělávacích systémech je běžnou praxí rozdělovat studenty do tříd a následně do menších skupin pro různé předměty a aktivity, neboť tento postup významně přispívá k efektivitě vzdělávacího procesu jak pro žáky, tak pro učitele. Dělbá studentů do menších skupin umožňuje přizpůsobit výuku individuálním potřebám žáků, což zvyšuje jejich zapojení a porozumění látce. Příkladně v jazykových předmětech je nezbytné vést aktivní konzultace, které jsou v menších skupinách realizovatelné mnohem efektivněji.

Dalším důvodem pro dělení žáků je kapacitní omezení učeben, které by při výuce v rámci celého třídního kolektivu nebylo možné optimálně využít. Tato opatření vedou k vyšší kvalitě výuky a efektivnějšímu využití dostupných zdrojů.

2.1 Definice

Class C : třída je množina všech studentů a je označena jako C . Formálně:

$$C = \{s_1, s_2, \dots, s_n\}$$

kde s_k je student a n je celkový počet studentů ve třídě.

SubClass $SC_{i/j}$: žákovská skupina (dále jen skupina) je podmnožina třídy C , kde i je číslo skupiny a j je počet částí. Formálně:

$$SC_{i/j} \subset C$$

2.2 Podmínky pro skupiny

Abychom zajistili, že žádný student není opomenut a že všechny skupiny jsou disjunktní, musí být splněny následující podmínky:

Úplnost

Sjednocení všech j částí skupiny musí tvořit celou třídu C :

$$\bigcup_{i=1}^j SC_{i/j} = C$$

Disjunkce

Žádný student nesmí patřit do více než jedné části téže skupiny:

$$SC_{i/j} \cap SC_{k/j} = \emptyset \quad \text{pro } i \neq k$$

2.3 Kategorizace podmnožin

V rámci tříd můžeme studenty rozdělit do různých kategorií podmnožin podle různých kritérií. Níže uvádíme několik obecných kategorií podmnožin a příklad jejich využití.

Dělení na zlomky

Jedním z nejběžnějších způsobů rozdělení třídy je dělení na zlomky. Tato metoda je často využívána pro různé aktivity a předměty, kde je potřeba pracovat s menšími skupinami studentů.

Příklad: Rozdělení třídy na dvě poloviny:

$$SC_{1/2} \quad \text{a} \quad SC_{2/2}$$

kde:

$$SC_{1/2} \cup SC_{2/2} = C$$

a

$$SC_{1/2} \cap SC_{2/2} = \emptyset$$

Tento přístup je užitečný například při laboratorních cvičeních, kde je potřeba, aby každý student měl přístup k vybavení a učitel mohl efektivně dohlížet na práci studentů.

Dělení podle pohlaví

Další běžnou metodou rozdělení studentů je podle pohlaví. Toto rozdělení může být užitečné v situacích, kde je potřeba řešit specifické potřeby studentů podle pohlaví.

Příklad: Rozdělení třídy na dívky a chlapce:

$$SC_{\text{dívky}} \quad \text{a} \quad SC_{\text{chlapci}}$$

kde:

$$SC_{\text{dívky}} \cup SC_{\text{chlapci}} = C$$

a

$$SC_{\text{dívky}} \cap SC_{\text{chlapci}} = \emptyset$$

Tento přístup může být užitečný při tělesné výchově nebo v předmětech, kde jsou specifické fyziologické nebo psychologické rozdíly mezi pohlavími relevantní.

2.4 Důsledky a diskuse

Úplnost

Podmínka úplnosti zajišťuje, že každý student ve třídě C je přiřazen do alespoň jedné skupiny. Pokud by některý student nebyl zahrnut v žádné skupině, nebyla by splněna podmínka:

$$\bigcup_{i=1}^j SC_{i/j} \neq C$$

To by znamenalo, že existuje alespoň jeden student s_k , který nepatří do žádné části skupiny, což je nepřijatelné.

Disjunkce

Podmínka disjunkce zajišťuje, že žádný student není přiřazen do více než jedné části skupiny. Pokud by některý student patřil do více než jedné části skupiny, byla by narušena disjunkčnost:

$$SC_{i/j} \cap SC_{k/j} \neq \emptyset \quad \text{pro } i \neq k$$

To by znamenalo, že existuje alespoň jeden student s_k , který patří do více než jedné části skupiny, což je také nepřijatelné.

2.5 Příklady

Třída a podskupiny

Předpokládejme, že máme třídu C se n studenty:

$$C = \{s_1, s_2, \dots, s_n\}$$

Dále předpokládejme, že třídu C chceme rozdělit do j disjunktních skupin:

$$\{SC_{1/j}, SC_{2/j}, \dots, SC_{j/j}\}$$

Úplnost

$$\bigcup_{i=1}^j SC_{i/j} = C$$

Tento výraz zajišťuje, že každá část skupiny je zahrnuta ve sjednocení, což pokrývá celou třídu.

Disjunkce

$$SC_{i/j} \cap SC_{k/j} = \emptyset \quad \text{pro } i \neq k$$

Tento výraz zajišťuje, že žádný student není zahrnut ve více než jedné části skupiny.

Příklad pro tři části

Pro ilustraci uvažujme třídu C se šesti studenty:

$$C = \{s_1, s_2, s_3, s_4, s_5, s_6\}$$

Chceme třídu rozdělit do tří disjunktních skupin:

$$\{SC_{1/3}, SC_{2/3}, SC_{3/3}\}$$

Předpokládejme, že:

$$SC_{1/3} = \{s_1, s_2\}$$

$$SC_{2/3} = \{s_3, s_4\}$$

$$SC_{3/3} = \{s_5, s_6\}$$

Poté platí:

$$SC_{1/3} \cup SC_{2/3} \cup SC_{3/3} = \{s_1, s_2, s_3, s_4, s_5, s_6\} = C$$

a

$$SC_{1/3} \cap SC_{2/3} = \emptyset$$

$$SC_{1/3} \cap SC_{3/3} = \emptyset$$

$$SC_{2/3} \cap SC_{3/3} = \emptyset$$

Závěr

Matematická úvaha o třídách a skupinách zajišťuje, že rozdělení studentů do skupin je úplné a disjunktní. To znamená, že každý student je přiřazen do alespoň jedné skupiny a žádný student není přiřazen do více než jedné části skupiny. Tímto způsobem je zajištěno, že žádný student není opomenut a že všechny skupiny jsou disjunktní a kompletně pokrývají celou třídu.

2.6 Implementace

V předchozí kapitole jsme diskutovali matematickou úvahu o třídách a jejich skupinách (SubClasses), kde jsme stanovili tři klíčové podmínky: úplnost, disjunkci a podmnožinovou strukturu. V této kapitole se zaměříme na implementaci těchto matematických principů do třídního a databázového modelu. Zároveň se pokusíme najít nejefektivnější řešení, které zajistí plnou konzistenci dat v databázi.

Třídní Model

Třídní model zahrnuje několik hlavních entit: `Class`, `SubClass`, `Student` a `StudentAssignment`. Model `Class` reprezentuje třídu jako celek, model `SubClass` reprezentuje skupiny v rámci třídy, model `Student` reprezentuje jednotlivé studenty a model `StudentAssignment` zaznamenává přiřazení studentů k třídám a skupinám.

Class

`Class` obsahuje informace o třídě, jako jsou identifikátor třídy, označení třídy, prefix, datum platnosti od a do, místnost a třídní učitel.

SubClass

`SubClass` reprezentuje skupiny studentů v rámci třídy. Každá skupina má své vlastní identifikační číslo, název a odkaz na třídu, ke které patří. `SubClass` může existovat pouze v kontextu třídy, což znamená, že každá `SubClass` musí být vždy přiřazena konkrétní třídě.

Student

`Student` obsahuje informace o jednotlivých studentech, jako jsou identifikační číslo, uživatelské jméno, jméno a příjmení.

StudentAssignment

`StudentAssignment` zaznamenává přiřazení studentů k třídám a skupinám. Obsahuje odkazy na studenty, třídy a skupiny, ke kterým jsou přiřazeni. Klíčovým prv-

kem tohoto modelu je zajištění trvanlivosti přiřazení, tedy schopnost zaznamenávat změny přiřazení studentů v průběhu roku.

V průběhu školního roku mohou nastat různé změny, které vyžadují aktualizaci přiřazení studentů. Tyto změny mohou zahrnovat mnoho faktorů a ty nejčastější z nich jsou vyjmenovány níže. Trvanlivost přiřazení umožňuje udržovat historický záznam o všech těchto změnách, což je nezbytné pro správu.

Problémy a důvody pro trvanlivost:

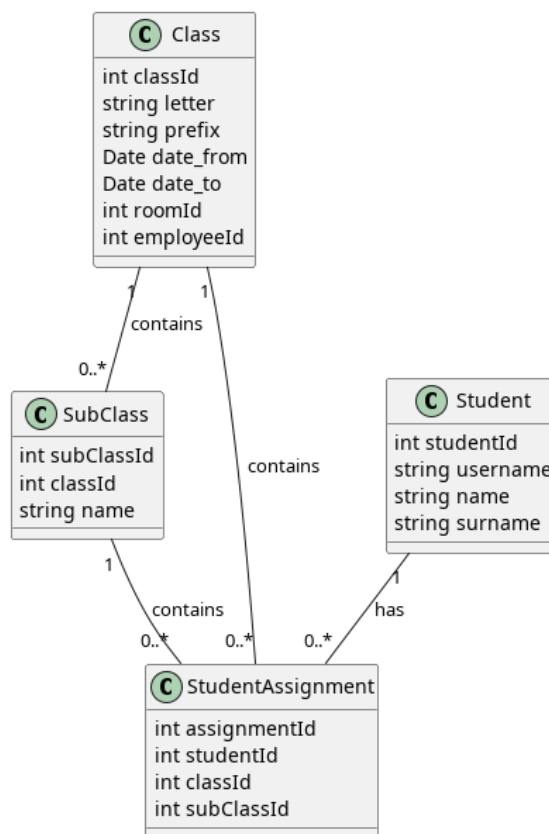
- **Přechody mezi třídami:** Studenti mohou být během školního roku přeřazeni do jiné třídy z různých důvodů, jako jsou změny nálad v třídním kolektivu nebo přestup studenta mezi obory.
- **Změny ve složení skupin:** V závislosti na specifických potřebách výuky mohou být studenti přeřazeni mezi různými skupinami. Například při výuce jazyků může být potřeba změnit složení skupin kvůli homogenizaci, nebo naopak heterogenizaci skupiny, čímž v závislosti na potřebách jednotlivých žáků lze docílit zefektivnění výuky.

Model `StudentAssignment` musí být schopen zaznamenávat všechny tyto změny, aby bylo možné udržovat konzistentní a aktuální přehled o přiřazení studentů. Tento model tedy umožňuje sledování přiřazení studentů k třídám a skupinám v určitém časovém období a zajišťuje, že historické změny jsou správně zaznamenány.

Schopnosti modelu:

- **Referenční integrita:** Každý záznam o přiřazení musí odkazovat na platné záznamy ve třídách a skupinách.
- **Flexibilita přiřazení:** Umožňuje přiřazení studenta buď přímo do třídy nebo do specifické skupiny.
- **Trvanlivost:** Udržuje historické záznamy o přiřazení, které umožňují sledovat změny v průběhu školního roku.

Tímto způsobem model `StudentAssignment` nejen zajišťuje aktuálnost a konzistenci dat, ale také poskytuje cenné historické informace, které mohou být využity pro analýzu a zlepšení vzdělávacího procesu.



Obrázek 2.1: Model tříd žákovských skupin

Databázový Model

Databázový model musí zajistit, že všechny podmínky úplnosti, disjunkce a podmnožinové struktury jsou dodrženy. Toho lze dosáhnout pomocí správného nastavení cizích klíčů, unikátních omezení a triggerů.

Úplnost

Úplnost zajišťuje, že každá třída je pokryta skupinami tak, aby žádný student nezůstal nezařazen. V aplikační logice zajistíme, že všechny skupiny dohromady pokrývají celou třídu. To můžeme kontrolovat při každém přiřazení studenta do skupiny. Je nezbytné, aby všechny skupiny třídy byly disjunktní a jejich sjednocení tvořilo celou třídu.

Disjunkce

Disjunkci zajistíme na úrovni databáze pomocí unikátních omezení. Každý student může patřit pouze do jedné skupiny určitého typu v dané třídě. To lze implementovat pomocí unikátních indexů nebo pomocí aplikační logiky, která před přiřazením studenta do skupiny provede potřebné kontroly.

Podmnožinová struktura

Podmnožinovou strukturu zajistíme pomocí cizích klíčů a referenční integrity. Každá skupina musí odkazovat na existující třídu a každé přiřazení studenta do skupiny musí odkazovat na existujícího studenta, třídu a skupinu.

Nejefektivnější řešení

Pro zajištění úplné konzistence dat v databázi je nutné implementovat následující mechanismy.

Validace na úrovni aplikace

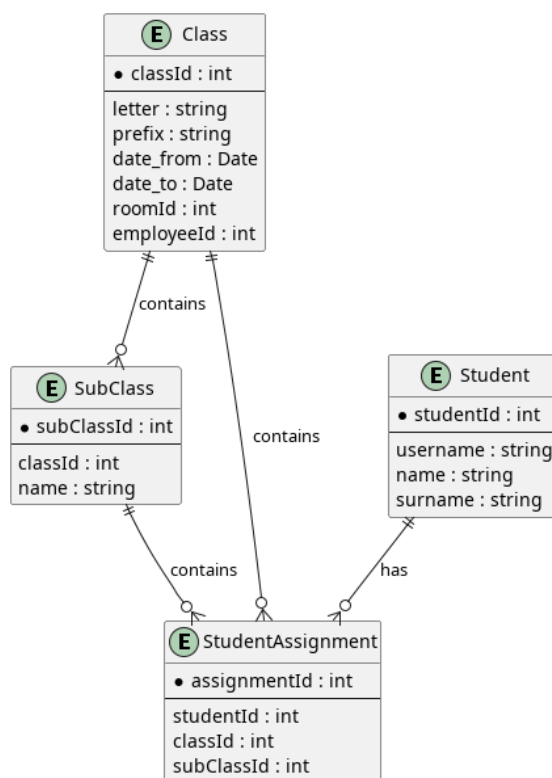
Při každém přiřazení studenta do skupiny zkontrolujeme, zda přiřazení splňuje podmínky úplnosti a disjunkce. Pokud podmínky nejsou splněny, přiřazení se neprovede a uživatel obdrží příslušnou chybovou zprávu. Aplikační logika bude muset zohlednit také případy, kdy se celá třída učí společně a není třeba přiřazovat studenty do skupin.

Unikátní omezení

Na úrovni databáze zavedeme unikátní omezení, která zajistí, že každý student může patřit pouze do jedné skupiny určitého typu v dané třídě. Tím zajistíme disjunkci dat.

Referenční integrita

Pomocí cizích klíčů zajistíme, že každá skupina je podmnožinou existující třídy a každé přiřazení studenta odkazuje na existujícího studenta, třídu a skupinu. Tím zajistíme podmnožinovou strukturu.



Obrázek 2.2: Referenční identita

Diskuse o efektivitě řešení

Implementace výše uvedených mechanismů v kombinaci s aplikační logikou poskytuje robustní řešení pro správu tříd a jejich skupin ve vzdělávacím systému. Validace na úrovni aplikace umožňuje flexibilitu a zajišťuje, že jsou splněny všechny matematické podmínky. Unikátní omezení a referenční integrita na úrovni databáze zajišťují konzistenci dat a minimalizují riziko chyb.

Tímto způsobem zajišťujeme:

- **Úplnost**: Každá třída je kompletně pokryta skupinami a žádný student nezůstane nezařazen.
- **Disjunkce**: Každý student může být přiřazen pouze do jedné skupiny určitého typu v dané třídě.
- **Podmnožinová struktura**: Každá skupina je vždy podmnožinou konkrétní třídy, ke které patří.

3 tridnice

4 Reference

- [1] ČESKO. § 31 odst. 1 zákona č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon) - znění od 1. 1. 2024. Revidováno: 17.06.2024. URL: <https://www.zakonyprolidi.cz/cs/2004-561#p66-5>.
- [2] ČESKO. § 31 odst. 1 zákona č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon) - znění od 1. 1. 2024. Revidováno: 17.06.2024. URL: <https://www.zakonyprolidi.cz/cs/2004-561#p66-6>.

5 Přílohy

- Analýza pro proces modernizace školního informačního systému pro střední školu, Daniel Adámek