

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Databázové systémy

Dokumentace k projektu

2. května 2022

Michal Šmahel (xsmah01)
Martin Havlík (xhavli56)

Obsah

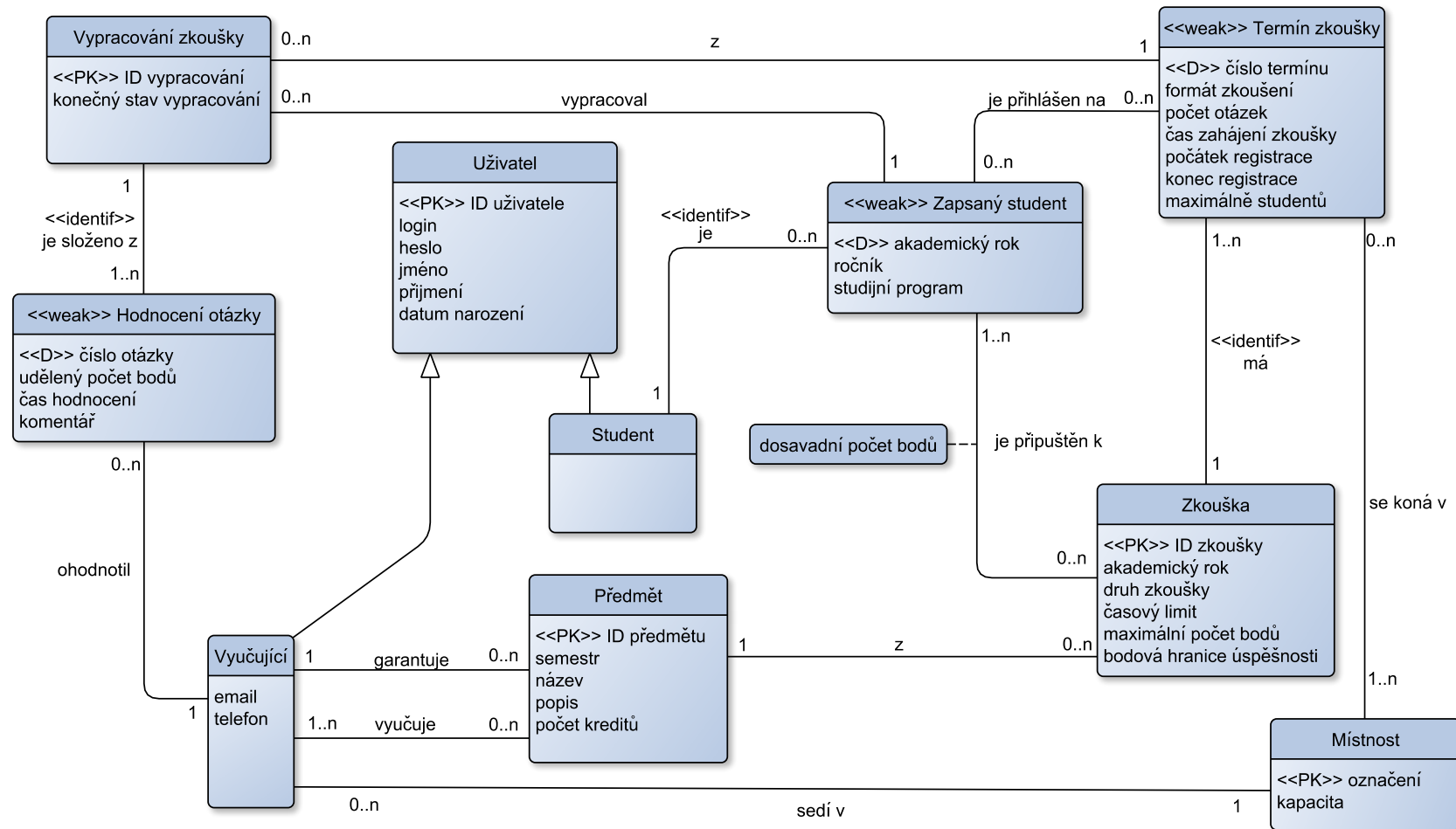
| | | |
|----------|---|-----------|
| 1 | Zadání přejaté z předmětu IUS | 2 |
| 2 | ER diagram (Entity Relationship Diagram) | 3 |
| 3 | Model případů užití (Use Case Diagram) | 6 |
| 4 | Logické schéma databáze | 8 |
| 4.1 | Diagram funkčních závislostí | 10 |
| 5 | Triggery, uložené procedury, pohledy | 11 |
| 5.1 | Triggery | 11 |
| 5.2 | Procedury | 11 |
| 6 | Přístupová práva | 11 |
| 7 | Optimalizace zpracování dotazů | 12 |

1 Zadání přejaté z předmětu IUS

Jelikož byla možnost převzít zadání z předmětu Úvod do softwarového inženýrství, projekt se zakládá na zadání Zkoušky (č. 20). Diagramy odevzdané v rámci předmětu IUS však byly významně pozměněny, aby lépe odpovídaly požadavkům projektu v předmětu Databázové systémy.

Původní zadání znělo následovně: „Navrhněte modul fakultního informačního systému, který bude umožňovat studentům se přihlašovat na zkoušky a vyučujícím tyto zkoušky hodnotit. Zkouška může být buď semestrální nebo půl semestrální, přičemž semestrální zkouška má vždy alespoň 3 termíny. Studenti se mohou na zkoušky přihlašovat pouze pokud je přihlašování otevřeno, u semestrální zkoušky musí navíc systém zajistit, že se student nesmí přihlásit na více než 3 termíny a také, že na další termín se může přihlásit až po proběhnutí termínu, na který je aktuálně přihlášen. Pro každou zkoušku je potřeba uchovávat základní informace o ní (datum a čas, kdy se koná, maximální počet studentů, jenž se mohou přihlásit, maximální počet bodů ze zkoušky a minimální počet bodů potřebný pro úspěšné složení zkoušky, počet otázek, ...). Body ze zkoušky mohou vkládat pouze učitelé, jenž jsou k předmětu, pod který zkouška spadá, přiřazeni. Pokud má zkouška nastaveno více otázek, musí hodnocení obsahovat body pro každou z těchto otázek. Hodnocení také může obsahovat komentář. Kromě přihlašování musí systém poskytnout studentům i možnost se odhlásit (pokud je přihlašování stále otevřeno). Studenti také mohou zjišťovat body získané ze zkoušky spolu s informacemi kdo a kdy tuto zkoušku hodnotil.“

2 ER diagram (Entity Relationship Diagram)



Projekt vychází ze zadání Zkoušky z předmětu Úvod do softwarového inženýrství (viz sekci 1). Řeší však i některé případy nad jeho rámec. Při řešení bylo usouzeno, že díky vlastnímu rozšíření vzniknou diagramy popisující situaci bližší realitě. Kromě toho nadstavba umožňuje použití některých prvků, které vyžaduje zadání projektu v předmětu Databázové systémy nad rámec předmětu Úvod do softwarového inženýrství. I přesto však zachycujeme pouze jeden modul teoretického informačního systému fakulty a bylo kvůli tomu třeba učinit několik ústupků.

V následujícím textu budou stručně popsány entitní a vztahové množiny z ER diagramu umístěného na straně 3. Některé entitní množiny vyžadují podrobnější popis kvůli lepšímu pochopení situace zachycené diagramem. Pro zvýšení přehlednosti jsou entitní množiny popsány v bodech seskupených pod názvy entitních množin.

Uživatel, Vyučující, Student

- Pro uchování dat o uživateli informačního systému.
- Jeden uživatel může být zároveň vyučujícím i studentem (doktorand).
- Role uživatele v systému se v čase může měnit (student se po absolvování stane vyučujícím).
- Atribut „heslo“ značí pouze otisk hesla.
- U vyučujícího jsou oproti studentovi uvedeny kontaktní údaje pro potřeby studentů či jiných vyučujících, přičemž uvedený email může být soukromý (nefakultní, neodvozený z loginu vyučujícího).

Zapsaný student

- Studenti se každý akademický rok zapisují do nového ročníku.
- Studijní program se může mezi akademickými roky měnit (např. bakalářský → magisterský).
- Silně závislé na studentovi (uživatel musí být studentem, aby se mohl zapsat).

Zkouška

- Vypsání zkoušky bez konkrétního termínu (obecné pojetí sdružující jednotlivé termíny).
- Může být půlsestrální nebo semestrální (atribut „druh zkoušky“).
- Ne všichni zapsaní studenti musí být ke zkoušce připuštěni, proto existuje vztahová množina „je připuštěn k“ mezi zapsaným studentem a zkouškou.
- Pro účely zjištění úspěšnosti u zkoušky je nutné znát počet bodů získaných v průběhu semestru před zkouškou (týká se zejména semestrálních zkoušek), proto má vztahová množina „je připuštěn k“ atribut „dosavadní počet bodů“. Minimum samotné zkoušky, pokud jej má, určuje atribut „bodová hranice úspěšnosti“.
- Zkouška je vždy z nějakého konkrétního předmětu.

Termín zkoušky

- Zkouška může mít několik termínů, na které se mohou přihlásit zapsaní studenti, kteří byli připuštěni ke zkoušce.
- V různých termínech se může zkoušet různým způsobem (např. písemně v běžném režimu a ústně v distanční formě v případě virové nákazy studenta) – „formát zkoušení“.

- Počet otázek se může u jednotlivých termínů lišit, i když v součtu vydají za stejné množství bodů.
- I přestože je počet otázek později odvoditelný z entitní množiny Hodnocení otázky, je nutné tento údaj znát již při prvním zadávání pro zajištění zadání správného počtu hodnocení otázek.
- Počet studentů nemusí být omezen jen kapacitou místností, ale také předpisy apod. – atribut „maximálně studentů“.

Vypracování zkoušky

- Vypracováním zkoušky se rozumí výstup práce zkoušeného studenta v daném formátu zkoušení.
- Jedná se o sdružení ohodnocených otázek pro daného studenta v daném termínu zkoušky.
- Atribut „konečný stav vypracování“ značí, zda byla zkouška vypracována řádně, student byl ze zkoušky vyloučen nebo ji nemohl dokončit z jiných vážných důvodů.

Hodnocení otázky

- K vypracování zkoušky se silně váže hodnocení jednotlivých otázek.
- Pro každou otázku je uveden počet bodů a nepovinný komentář.

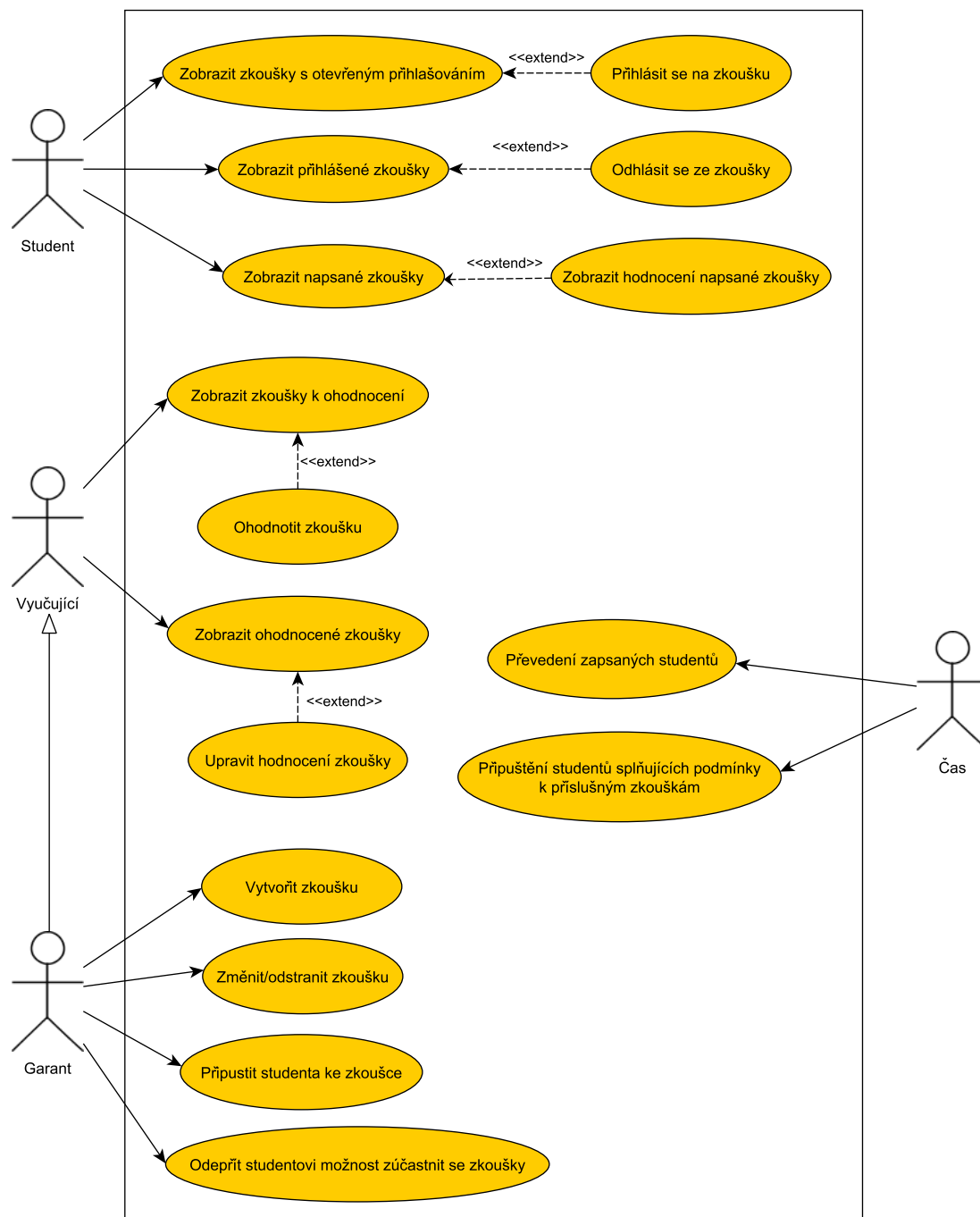
Předmět

- Předmět není přímo vázán na studenty, jelikož pro účely zkoušek není vyloženě nutné vědět, jaké předměty studenti aktuálně studují. Toto řeší jiné moduly informačního systému.
- Předmět není identifikován zkratkou, ale generovaným identifikátorem, aby bylo možné měnit garanta nebo počet kreditů a zachovat historii (např. kvůli pozdějším statistikám). V případě změny se vloží nový záznam předmětu a starý se přestane používat.
- Popis je pouze stručné shrnutí informací o předmětu kvůli přívětivější identifikaci předmětů v rámci zkouškového modulu.
- Předmět má jednoho garanta a alespoň jednoho vyučujícího, přičemž do vyučujících spadají jak přednášející, tak i cvičící daného předmětu. Zástupce garanta není uvažován.

Místnost

- Místnost značí kancelář nebo přednáškovou aulu, laboratoř a jiné typy místností, kde se dají psát zkoušky.
- Potenciální zkušební místnosti potřebují mít uvedenou kapacitu, kterou je možné využít při plánování zkoušek.

3 Model případů užití (Use Case Diagram)



Následující text se věnuje doplnění informací k diagramu případů užití ze strany 6. Budou popsány akce jednotlivých aktérů, avšak v rámci stručnosti nebudou popisovány jednotlivé případy užití, ale možnosti daného aktéra jako celek. Některé části textu se odkazují na informace zmíněné u ER diagramu.

Student

- Student si může zobrazit zkoušky, k nimž byl připuštěn, a mají otevřené přihlašování. Následně se může přihlásit na některý z volných termínů.
- Může si zobrazit přihlášené zkoušky a případně se z nich odhlásit.
- Dále se dostane k proběhlým zkouškám, na něž byl přihlášen. U jednotlivých zkoušek si může zobrazit jejich hodnocení, pokud již bylo zadáno do systému.

Vyučující

- Vyučující má přístup ke zkouškám, z jím vyučovaných předmětů, které je třeba opravit. Kromě jejich zobrazení má právo na zadávání konkrétních hodnocení.
- Obdobně si může zobrazit již ohodnocené zkoušky, k nimž má přístup a upravit dané hodnocení (pro účel oprav).

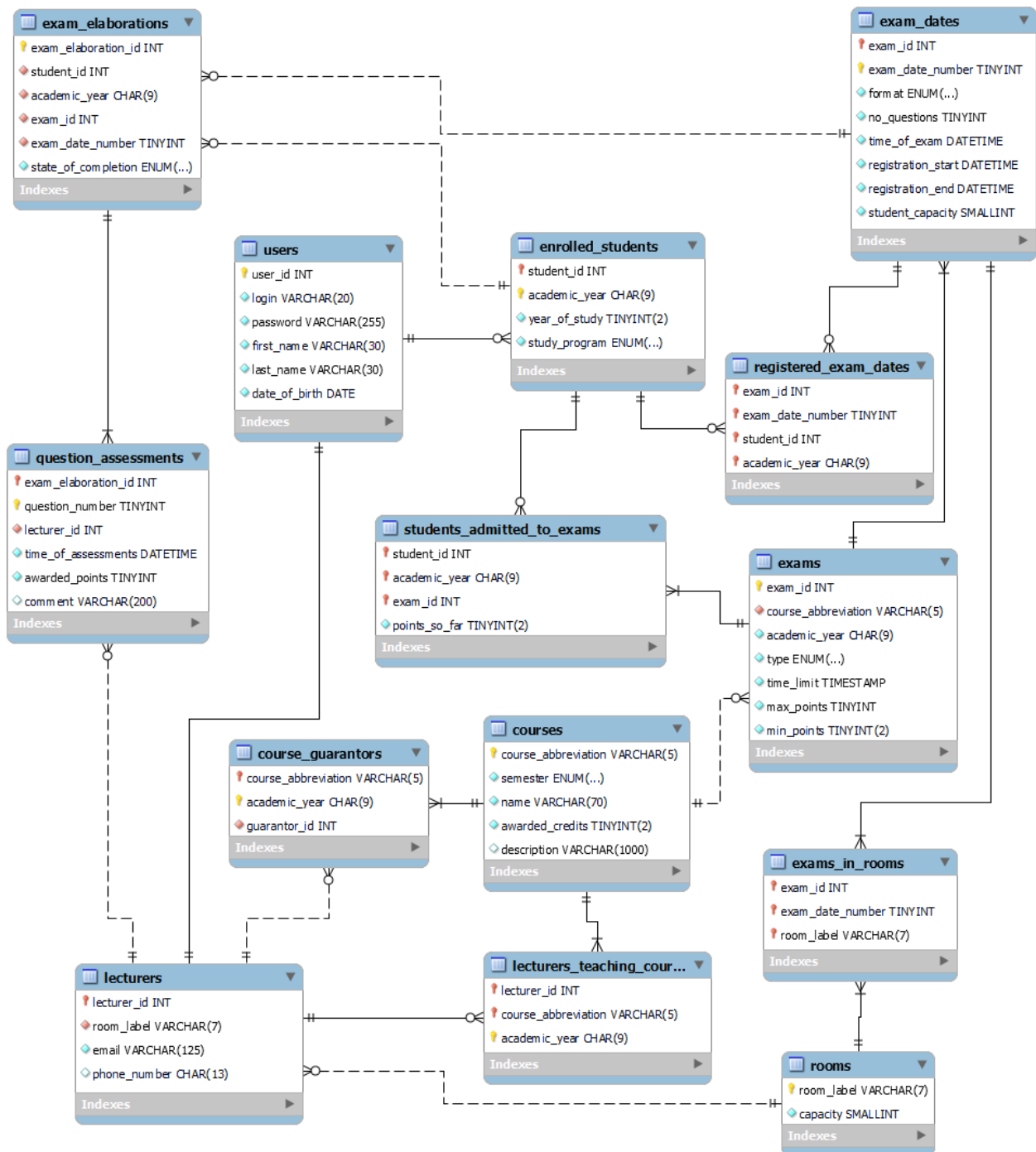
Garant

- Garant, jako speciální případ vyučujícího, má navíc oprávnění vytvářet a případně administrovat zkoušky (měnit je a odstraňovat).
- V případě potřeby může dodatečně připustit konkrétního studenta k dané zkoušce (pokud mu byl udělen zápočet po ukončení zápočtového týdne z důvodu doložené nemoci apod.).
- V případech, kdy se zjistí podvodné chování studenta v postižitelné míře, má garant právo studentovi odeprít možnost zúčastnit se zkoušky (akce může být navázaná na odebrání zápočtu v jiném modulu).

Čas

- Speciálním účastníkem je čas, který zastupuje automatické chování informačního systému.
- Stará se o převádění zapsaných studentů z odpovídajícího modulu do zkouškového modulu po ukončení zápisů.
- Po ukončení zápočtového týdne připouští studenty splňující potřebné podmínky k příslušným zkouškám. Seznam studentů přebírá od modulu, který řeší průběžné hodnocení studentů během semestru.
- V případě půl semestrální zkoušky může být termín připuštění a potřebné podmínky nastaveny tak, aby byli připuštěni všichni studenti např. již na začátku semestru.

4 Logické schéma databáze



Uvedené logické schéma databáze bylo modelováno v programu MySQL Workbench¹. Datové typy jednotlivých atributů jsou tedy rozdílné oproti databázi Oracle a slouží tedy pouze k orientačním účelům. Legenda k symbolům před atributy je k nalezení zde: <https://stackoverflow.com/a/28859886>.

Při převodu ER diagramu na schéma databáze byly učiněny určité změny či dodatky, zejména za zmínku stojí:

Zapsaný student → enrolled_students

- Z původního návrhu, kdy student byl specializací uživatele a vázala se k němu slabá entitní množina Zapsaný student, došlo ke zjednodušení, ve kterém je student modelován pouze více „samostatnými“ záznamy enrolled_students.
- Enrolled_students tedy zastává roli oné specializace uživatele jako studenta.

Zkratka předmětu

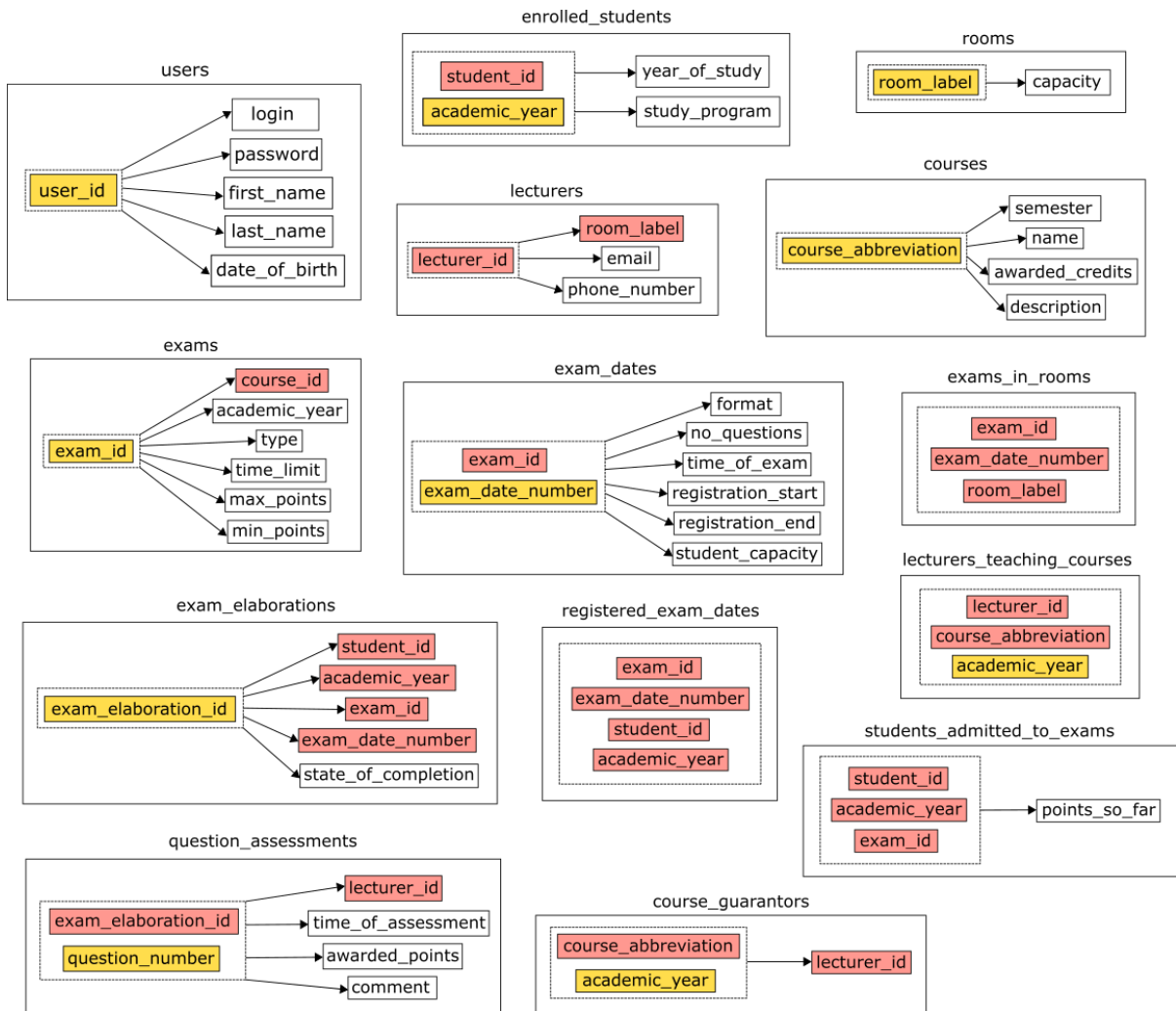
- Nevýznamový primární klíč v podobě ID byl nahrazen zkratkou předmětu, s předpokladem její jedinečnosti.

Vyučující vyučuje, garantuje

- Zmíněné dva vztahy byly povýšeny na vztahové množiny navíc obsahující akademický rok, kdy daný vyučující zastával tu kterou roli.

¹<https://www.mysql.com/products/workbench/>

4.1 Diagram funkčních závislostí



V rámci tvorby schématu databáze byl také vytvořen diagram funkčních závislostí. Pro primární a cizí klíče využívá barevné značení obdobné MySQL Workbench, viz legendu odkazovanou z kapitoly 4.

Snad jediná závislost, která byla při návrhu dosti diskutována a mohla by porušovat BCNF se nachází v entitní množině `exam_elaborations`. Jako primární klíč by se nabízel složený klíč z atributů `student_id`, `academic_year`, `exam_id` a `exam_date_number`, jelikož tato čtveřice jednoznačně identifikuje danou vypracovanou zkoušku. Poté by `state_of_completion` závisel na takto složeném primárním klíči.

Avšak pro jednodušší přístup a práci s tabulkou v dotazech byl vytvořen separátní, jediný primární klíč v podobě generovaného identifikátoru (ID). Takto nyní ostatní neklíčové atributy závisí na ID vypracování zkoušky: určitou zkoušku v určitém termínu (`exam_date_number`) napsal daný student, v daném akademickém roce a dané vypracování je buď úspěšně dokončené, nedokončené či student byl ze zkoušky vyloučen – možné hodnoty `state_of_completion`.

Při takovémto úhlu pohledu na entitní množinu `exam_elaborations` a význam jednotlivých závislostí je BCNF i zde zachována.

5 Triggery, uložené procedury, pohledy

5.1 Triggery

Pro generování sekvencí pořadových čísel otázek v jednotlivých vypracováních zkoušky a termínů jednotlivých zkoušek (pro sloupce, které vznikly z diskriminátorů slabých entitních množin) byly vytvořeny triggery `bi_tg_question_assessments_dk` a `bi_tg_exam_dates_dk` reagující na akci `BEFORE INSERT`. Tyto triggery zjistí maximální existující číslo otázky/termínu a nastaví číslo nově vkládaného záznamu o jedna větší.

Další trigger, `bi_tg_maximum_registered_exam_dates`, zajišťuje, aby nebyl překročen maximální povolený počet zaregistrovaných termínů u zkoušek: 3 termíny pro semestrální a 1 termín pro půlsemestrální zkoušku. Spouští se s akcí `BEFORE INSERT`, využívá uživatelsky definované výjimky a v případě nepovolené registrace vyvolává aplikační chybu.

5.2 Procedury

K registraci studenta na termín zkoušky slouží procedura `usp_register_exam_date`, která zprvu kontroluje určité aspekty registrace, např. zda je student vůbec ke zkoušce připuštěn (přeloženo jako „admitted“), zda je registrace spuštěna a ještě není ukončena apod. V případě splnění požadovaných kontrol jsou nová data vložena do tabulky `registered_exam_dates`. Zamýšlené použití této procedury je pro registrace vykonávané např. garantem či plošně.

Varianta výše popsané procedury, tentokrát určená pro registraci termínu zkoušky studentem samotným, je `usp_register_my_own_exam_date`. Registruje studenta identifikovaného aktuálním uživatelem, jehož login je zjištěn pomocí `SYS_CONTEXT('USERENV', 'CURRENT_USER')` (předpokládáme kompatibilitu uživatelských loginů na databázovém serveru s navrhovanou databází)².

Druhá hlavní procedura je `usp_zero_out_results_if_zero_question`, která ve výsledcích daného termínu zkoušky nuluje pomocí kurzoru hodnocení za jednotlivé otázky, pokud alespoň jedna otázka je ohodnocena 0 body. Tímto realizuje podmínku „žádná podčást za nula“ vyskytující se u některých zkoušek.

6 Přístupová práva

Vedoucí projektu, Michal Šmahel (xsmah01), zaujal „roli“ admina či garanta systému a druhý člen týmu, Martin Havlík (xhavli56), zastal pozici studenta. Je tak možné jednoduše vymezit informace, ke kterým student nesmí mít přístup (např. detaily o uživateli, zda jsou ostatní studenti připuštěni k/zaregistrování na zkoušku či hodnocení ostatních studentů) a naopak k čemu student přístup má, jako stav jeho registrací, hodnocení, známky apod.

Kromě udělení práv pro určité tabulky byly také vytvořeny pohledy `my_available_exam_dates`, `my_exam_dates`, `my_marks` a `academics`³, ke kterým byla přidělena přístupová práva studentovi pro `SELECT`. Student si tak může zjistit termíny, na které se může přihlásit, na které je již přihlášen, souhrnný přehled jeho známek a informace o akademických.

Nad tabulkami zpřístupněnými studentovi a s využitím pohledu `academics` byl studentem vytvořen materializovaný pohled `courses_info` se souhrnem informací o jednotlivých kurzech.

²Procedura je součástí modelu přístupových práv popsaného v sekci 6.

³Nahrazuje přístup k tabulkám `users` a `lecturers`, kam by studenti neměli mít plný přístup.

7 Optimalizace zpracování dotazů

Pro EXPLAIN PLAN byl zvolen dotaz ze 3. části projektu, a to průměrné hodnocení jednotlivých termínů dané zkoušky.

```
WITH summary_results AS (  
SELECT exam_date_number, student_id, SUM(qa.awarded_points) points  
FROM enrolled_students  
JOIN registered_exam_dates USING (student_id, academic_year)  
JOIN exam_dates USING (exam_id, exam_date_number)  
JOIN exam_elaborations USING(student_id, academic_year, exam_id, exam_date_number)  
JOIN question_assessments USING(exam_elaboration_id)  
WHERE exam_id = 62  
GROUP BY exam_date_number, student_id  
)  
SELECT exam_date_number, AVG(points) points_average  
FROM summary_results  
GROUP BY exam_date_number;
```

Prvotní exekuční plán bez žádného explicitně vytvořeného indexu vypadá následovně:

| Id | Operation | Name |
|-----|-------------------|--------------------------|
| 0 | SELECT STATEMENT | |
| 1 | HASH GROUP BY | |
| 2 | VIEW | |
| 3 | HASH GROUP BY | |
| * 4 | HASH JOIN | |
| 5 | NESTED LOOPS | |
| * 6 | TABLE ACCESS FULL | EXAM_ELABORATIONS |
| * 7 | INDEX UNIQUE SCAN | PK_REGISTERED_EXAM_DATES |
| 8 | TABLE ACCESS FULL | QUESTION_ASSESSMENTS |

Komentář k některým operacím

- 2 – WITH AS je, vcelku logicky, zpracováváno jako pohled
- 6 – predicate information: `filter("EL"."EXAM_ID"=62)`, prochází se celá tabulka jen za účelem podmínky WHERE, což je jistě místo pro přímočarou optimalizaci indexem
- 7 – unique index vytvořen pro PK implicitně
- 8 – proběhne TABLE ACCESS FULL, jelikož se pro spojení využívá pouze část složeného primárního klíče, tedy implicitně vytvořený index zde nepomůže, oproti 7

Pro optimalizaci přístupu byl vytvořen index nad sloupcem `exam_id` v tabulce `exam_elaborations`. Poté nabyl exekuční plán této podoby:

| Id | Operation | Name |
|-----|-------------------------------------|------------------------------|
| 0 | SELECT STATEMENT | |
| 1 | HASH GROUP BY | |
| 2 | VIEW | |
| 3 | HASH GROUP BY | |
| * 4 | HASH JOIN | |
| 5 | NESTED LOOPS | |
| 6 | TABLE ACCESS BY INDEX ROWID BATCHED | EXAM_ELABORATIONS |
| * 7 | INDEX RANGE SCAN | IX_EXAM_ELABORATIONS_EXAM_ID |
| * 8 | INDEX UNIQUE SCAN | PK_REGISTERED_EXAM_DATES |
| 9 | TABLE ACCESS FULL | QUESTION_ASSESSMENTS |

Komentář k optimalizaci

- 6, 7: vidíme využití indexu, kde místo původního `TABLE ACCESS FULL` je využit `INDEX RANGE SCAN`

Vytvoření dalšího indexu nad sloupcem `exam_elaboration_id` v tabulce `question_assessments` navíc s definováním `DESC` vede na nahrazení `TABLE ACCESS FULL` za následující posloupnost operací:

| | | |
|----|-------------------------------------|----------------------------|
| 9 | VIEW | VW_GBC_11 |
| 10 | HASH GROUP BY | |
| 11 | TABLE ACCESS BY INDEX ROWID BATCHED | QUESTION_ASSESSMENTS |
| 12 | INDEX FULL SCAN | IX_QUESTION_ASSESSMENTS_ID |

Optimalizace pomocí pohledu nad `GROUP BY` je zajímavá, avšak využití indexu je nesporné a mělo by tedy vést k rychlejšímu zpracování.