

Slovenská technická univerzita

Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Databázové systémy

Projekt

Školská databáza

Adam Žúrek

Cvičiaci: Ing. Matej Kloska
Študijný odbor: Informatika
Ročník: 2. Bc
Akademický rok: 2016/2017

1. Zadanie

Vo vami zvolenom prostredí vytvorte databázovú aplikáciu, ktorá komplexne rieši nižšie definované scenáre vo vami zvolenej doméne. Presný rozsah a konkretizáciu scenárov si dohodnete s Vaším cvičiacim na cvičení.

- Zobrazenie prehľadu všetkých záznamov (rozumne stránkovaných, ak je to potrebné), kde v každom riadku figuruje aj číslo, pre ktorého výpočet sa využíva GROUP BY
- extend podscenár
- Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom
- Zobrazenie konkrétneho záznamu – detail
- Vytvorenie nového záznamu
- Aktualizácia existujúceho záznamu
- Vymazanie záznamu

Aplikácia musí mať grafické rozhranie. Je dôležité, aby scenáre boli realizované realisticky a naozaj poskytovali funkcionality tak, ako by ju očakával zákazník v danej doméne.

Scenáre, ktoré menia dáta, musia byť realizované s použitím transakcií (explicitné transakcie, autocommit) a aspoň jeden z nich musí zahŕňať prácu s viacerými tabuľkami.

2. Špecifikácia scenárov

Aplikácia obsahuje nasledujúce scenáre:

- **Vyhľadávanie prvkov v database**

A rectangular search input field with a light gray border. Inside the field, the text "Search..." is displayed in a light gray font. To the right of the text, there is a magnifying glass icon, also in light gray.

Obr. 1 - Realizuje sa pomocou ILIKE funkcie v SQL. Vyhľadávajú sa študenti a učitelia podľa mena a priezviska.

- **Stránkovanie (po 8 prvkov)**



Obr. 2 - Funkčné je stránkovanie záznamov v tabuľke dopredu aj dozadu s funkciou navoliť si presné poradie stránky v tabuľke.

- **Pridávanie a upravovanie existujúceho prvku z tabuľky**

+ Add

Name	Surname	Scholarships	Actions
Matthew	Rodriguez	0	i ✎ ✖

Obr. 3 – Pridávanie a upravovanie prvku v tabuľke sa realizuje v samostatnom linku, do ktorého sa dostaneme pri pridávaní cez tlačidlo add a pri upravovaní cez tlačidlo v strede napravo.

name

Surname

+ Add

Obr. 4 – Samostatný link na vyplnenie pridania prvku alebo následného upravenia. Ošetrovanie zlého vstupu je taktiež implementované.

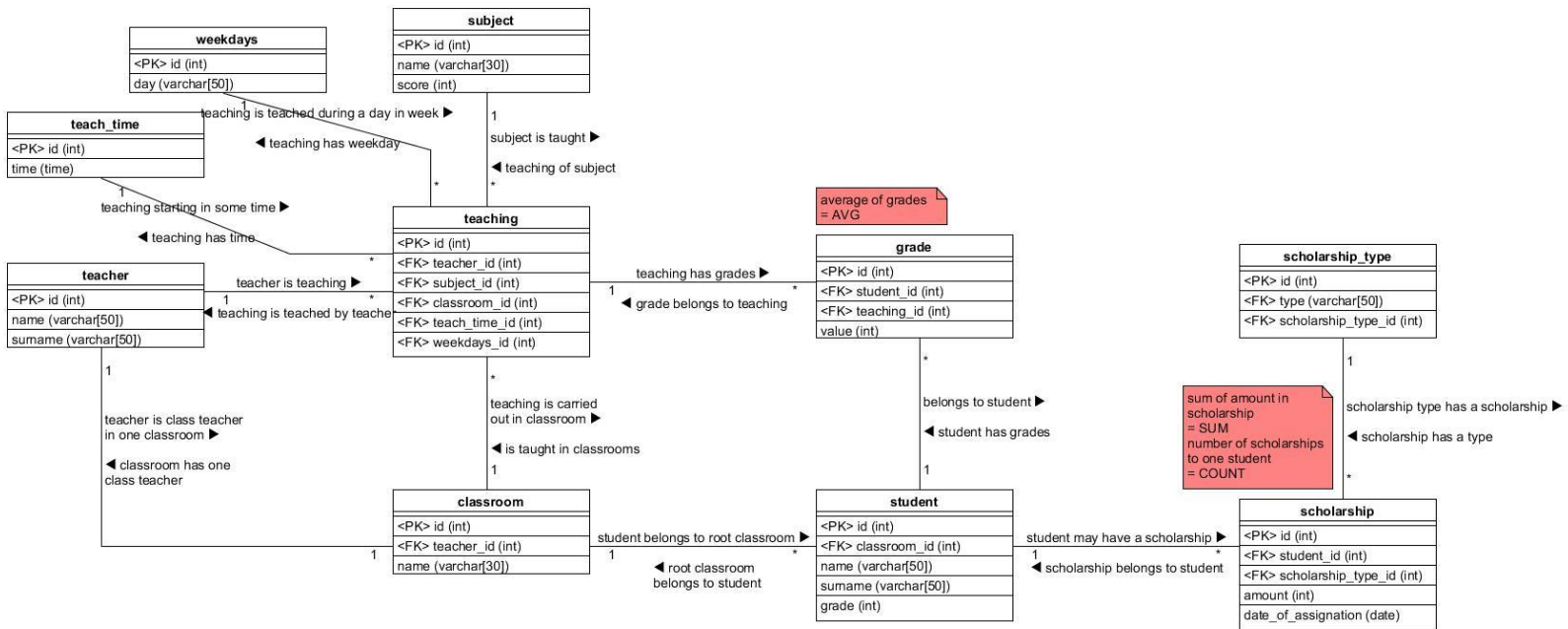
- **Vymazanie existujúceho prvku v tabuľke (za pomoci transakcie)**

Vymazanie sa realizuje cez červené tlačidlo napravo v obr. 3.

- **Zobrazenie detailných informácií o prvku v tabuľke, pričom sa používajú (agregačné funkcie, JOIN, GROUP BY, subselect)**

K detailným informáciám sa používateľ dostane cez prvé tlačidlo napravo s ikonkou informácií.

3. Diagram fyzického dátového modelu



Obr. 5 – DM diagram v. 2.0.

Dátový model je od predchádzajúceho odovzdania mierne upravený. Pribudli 3 číselníky a upravil som implementáciu tabuľky teaching, kde time_start (time) a time_end (time) som nahradil číselníkom časov a weekday (int) som nahradil číselníkom taktiež. V scholarship tabuľke som napojil typy štipendií taktiež číselníky. Vzťahy medzi tabuľkami sú opísané priamo v DM diagrame. Ak by obr.1 nebol dostatočne čitateľný, nachádza sa v projekte v súbore schema taktiež s jeho predchádzajúcou verziou.

3.1. Počet prvkov v jednotlivých tabuľkách

Students – 1 000 000

Teachers – 1 000

Teaching – 2 000 000

Grades – 2 000 000

Scholarships – 200 000

Classrooms – 100 000

4. Stručný opis návrhu a implementácie

Aplikácia slúži školám so správou študentov a zamestnancov. Informuje ich taktiež o ich vyučovaní, dôležitých parametroch vyučovania aj známkach.

4.1. Potrebne program k spusteniu aplikácie

Aplikácia je naprogramovaná:

- Frontend - HTML, Twig template engine, CSS, SASS, Java Script, angularJS
- Backend – PHP, Slim framework, PostgreSQL

Aplikácia na jej spustenie potrebuje:

- APACHE server
- PostgreSQL databáza
- Package manažéry – composer (PHP), bower (JS a CSS)
- Webový prehliadač

4.2. Programovacie prostredie

Pri návrhu aplikácie som využil programovacie prostredie PHPStorm od JetBrains.

4.3. Návrhové rozhodnutia

Program je štrukturovaný MVC návrhovým vzorom. Pripájanie na databázu sa realizuje pomocou PDO. V súbore .env sa nachádzajú nastavenia na pripojenie na databázu.

4.4. Opis implementácie jednotlivých scenárov

```
//filling table with students
$sql = "SELECT
    student.id, student.name, student.surname, count(scholarship.id) AS count_scholarship
FROM student
LEFT JOIN scholarship ON (student.id=scholarship.student_id)
";

//search students by name and surname
$values = array();
if (!empty($_GET['search'])) {
    $sql .= "WHERE coalesce(name, '') || ' ' || coalesce(surname, '') ILIKE ? ";
    $values[] = "%" . $_GET['search'] . "%";
}

//paging of students table
$_GET['offset'] = (empty($_GET['offset']) || !filter_var($_GET['offset'], FILTER_VALIDATE_INT) || $_GET['offset'] < 0) ? 0 : $_GET['offset'];
$sql .= "GROUP BY student.id LIMIT 8 OFFSET " . $_GET['offset'] * 8;
```

Obr. 6 – Vyhľadavanie prvku a stránkovanie v databáze.

```

);

if (empty($id)) { //add new student to the table
    $sql = "INSERT INTO student (name, surname) VALUES (?, ?)";
    App::log($sql);
}
else { //edit student in the table
    $values[] = $id;
    $sql = "UPDATE student SET name = ?, surname = ? WHERE id = ?";
    App::log($sql);
}

```

Obr. 7 – Pridávanie a upravovanie prvku v databáze.

```

static public function indexDelete($id){
    $conn = new Database();

    try { //transaction included
        $conn->pdoObject->beginTransaction();
        $stmt = $conn->prepare( query: "DELETE FROM grade WHERE student_id = ?");
        App::log( log: "DELETE FROM grade WHERE student_id = ?");
        $stmt->execute($conn->execute([$id]));
        $stmt = $conn->prepare( query: "DELETE FROM scholarship WHERE student_id = ?");
        App::log( log: "DELETE FROM scholarship WHERE student_id = ?");
        $stmt->execute($conn->execute([$id]));
        $stmt = $conn->prepare( query: "DELETE FROM student WHERE id = ?");
        App::log( log: "DELETE FROM student WHERE id = ?");
        $stmt->execute($conn->execute([$id]));
        $conn->pdoObject->commit();
    }
    catch (Exception $e) {
        $conn->pdoObject->rollBack();
    }
}

```

Obr. 8 – Mazanie prvku z tabuľky za pomoci transakcie.

```

static public function indexDetail($id){
    $conn = new Database();
    $sql = "SELECT
        coalesce(student.name, '') || ' ' || coalesce(student.surname, '') AS student, student.grade AS schoolyear, classroom.id AS classroom_id, classroom.name AS classroom,
        coalesce(teacher.name, '') || ' ' || coalesce(teacher.surname, '') AS teacher,
        (SELECT avg(value) FROM grade WHERE student_id = student.id) AS avg_grade,
        (SELECT sum(amount) FROM scholarship WHERE student_id = student.id) AS sum_scholarship
    FROM student
    LEFT JOIN classroom ON (student.classroom_id=classroom.id)
    LEFT JOIN teacher ON (classroom.teacher_id=teacher.id)
    WHERE student.id = ?
    ";
}

```

Obr. 9 – Detail na základné informácie o študentovi.

```
$sql = "SELECT
    subject.name AS subject, coalesce(teacher.name, '') || ' ' || coalesce(teacher.surname, '') AS teacher, classroom.name AS classroom
FROM teaching
LEFT JOIN classroom ON (teaching.classroom_id=classroom.id)
LEFT JOIN teacher ON (teaching.teacher_id=teacher.id)
LEFT JOIN subject ON (teaching.subject_id=subject.id)
WHERE teaching.classroom_id = ?
";
$stmt = $conn->prepare($sql);
```

Obr. 10 – Detail na základné informácie o vyučovaniach študenta.