

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Курсов проект

по

БАЗИ ОТ ДАННИ

СПЕЦ. ИНФОРМАТИКА, 3 КУРС, ЛЕТЕН СЕМЕСТЪР,

УЧЕБНА ГОДИНА 2018/19

ТЕМА: РЕАЛИЗАЦИЯ НА БАЗА ОТ ДАННИ ЗА
УЧИЛИЩЕ

29 май 2019 г.

Изготвил:

София

Иво Алексеев Стратев

Фак. номер: 45342

Група: 3

Съдържание

1	Увод (философски възглед)	3
2	Описание на множествата същности	3
2.1	Teacher	3
2.2	Class	3
2.3	Course	4
2.4	Program_entry	4
2.5	Exam	4
2.6	Parent	4
2.7	Student	4
2.8	Exam_Result	4
2.9	Връзки	5
3	Е/Р диаграма на модела на Базата данни	7
4	Преобразуване на Е/Р диаграмата до релационни схеми	9
4.1	Ограничения по реферетна цялост	10
5	Описание на функциите	12
5.1	full_gender	12
5.2	full_semester	12
5.3	day_of_week_as_string	12
6	Описание на тригерите	12
6.1	Before_Insert_Exam_Result	12
6.2	Before_Update_Exam_Result	12
6.3	Before_Insert_Course	12
6.4	Before_Update_Course	13
6.5	Before_Insert_Program_entry	13
6.6	Before_Update_Program_entry	13
7	Описание на изгледите	13
7.1	Grade	13
7.2	Program	13
8	Описание на приложението за достъп с резултатите от заявката	14
8.1	Резултати при изпълнение с combined_sql/test.sql	14
8.1.1	Създаване на таблица	14
8.1.2	Вмъкване в таблица	14

8.1.3	Ред, резултат от заявка към Grade	14
8.1.4	Ред, резултат от заявка към Program	14

1 Увод (философски възглед)

В типична база от данни описваща училище се описват същности - учител, ученик, родител. Въпреки, че те споделят общи характеристики като: собствено име, фамилно, адрес, електронна поща, мобилен и други. Тези характеристики биха се използвали по различен начин. Например по коренно различен начин бихме написали електронно писмо до всеки. По различен начин бихме говорили със всеки и всеки от тях би говорил с другия по различен начин. Тоест макар да споделят общи характеристики те биха се ползвали по различен начин в рамките на институцията училище. Наличието на общи характеристики е следствие на социалният елемент. Поради тази причина в текущата реализация се избягва използването на наследяване (is a) йерархия. Защото ако имахме общ родител например човек за тези същности, то това би било грешка в дизайна, която отразява реалността, но не отразява специфичното за областта знание, за липса на еднотипност на общите характеристики. По същия начин в текущата реализация на база данни се прави ясно разграничение между учебен час и час на класа. Тоест в текущата реализация е дадено предимство на спецификата на данните, вместо спазването на по-общото отразяване на действителността. В рамките на обектно ориентирания подход автора би използвал интерфейс за оказване на наличието на общи характеристики, които могат да бъдат извлечени, вместо общ родителски клас. С цел подчертаване на ясното разграничаване между същностите.

2 Описание на множествата същности

2.1 Teacher

Описва учител. Всеки учител има `id` - уникален идентификатор, `first_name` - собствено име, `last_name` - фамилно име, `gender` - пол, `work_phone` - служебен номер на мобилен, `email` - служебен адрес на електронна поща и `address` - личен адрес.

2.2 Class

Описва клас в училището. Всеки клас има `id` - уникален идентификатор, `school_year` - година на класа, с цел различаване на класовете през годините (например 12 Б клас през 2016 и 12 Б клас през 2014), `number` - номер на класа, `letter` - буква на класа за отличаване на паралелките, `class_room` - класна стая, в която се провежда часа на класа, `class_hour` - началото на часа на класа, приемаме, че часа на класа винаги е в петък.

2.3 Course

Описва предмет в училището. Всеки предмет има `id` - уникален идентификатор, на един конкретен клас се води само от един учител, `name` - име на предмета, `semester` - учебен срок на предмета.

2.4 Program_entry

Описва елемент в учебната програма. Всеки елемент на учебната програма има `id` - уникален идентификатор, `course` - предмет, който се преподава, `day_of_week` - в кой ден от седмицата се преподава, `room` - стая, в която се преподава, `starts` - кога започва съответния час, възможно е часовете за започват през различно време, а не на кръгъл час или на 15/20-та минута от кръглия час ..., `hours` - продължителност, брой учебни часове, допускаме преподаване на блокове.

2.5 Exam

Описва контролно/изпит/класно. Всяко контролно има `id` - уникален идентификатор, `course` - предмет на контролното, `room` - стая, в която се провежда, `starts` - време на започване, `hours` - брой учебни часове.

2.6 Parent

Описва родител/настойник на ученик. Всеки родител има `id` - уникален идентификатор, `first_name` - собствено име, `last_name` - фамилно име, `gender` - пол, `phone` - мобилен номер, `email` - адрес на електронна поща, `birthdate` - рождenna дата, `address` - основен адрес, `work` - професия.

2.7 Student

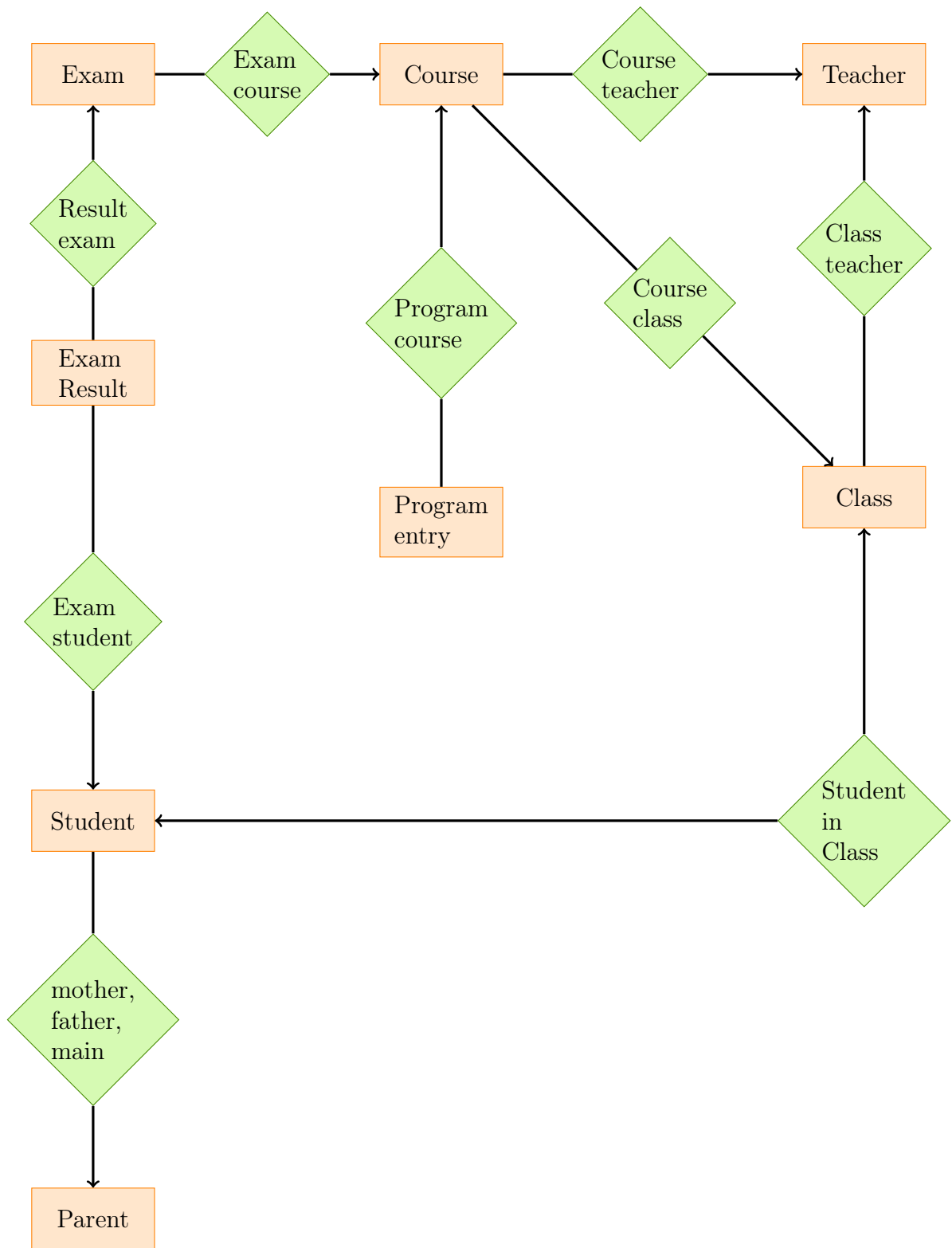
Описва ученик на ученик. Всеки ученик има `id` - уникален идентификатор, `mother` - майка, `father` - баща, `main_parent` - основен родител за контакт, `first_name` - собствено име, `last_name` - фамилно име, `gender` - пол, `phone` - мобилен номер, `email` - адрес на електронна поща, `birthdate` - рождenna дата, `address` - основен адрес.

2.8 Exam_Result

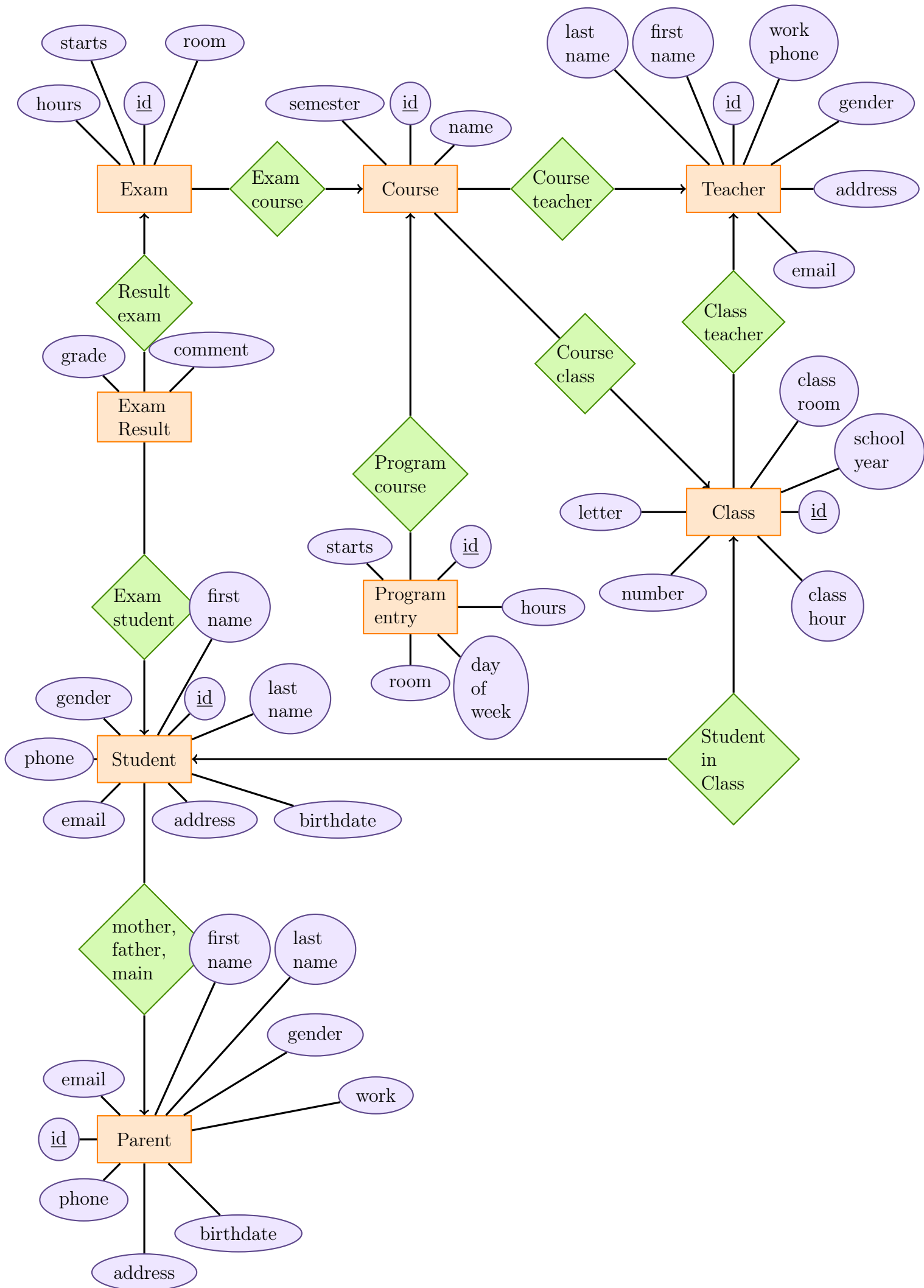
Описва оценка от контролна работа. Всяка оценка има уникален идентификатор - `id`, `grade` - съответната оценка, `comment` - коментар или обратна връзка към оценката.

2.9 Връзки

Всеки клас има точно един Class teacher (lead_teacher) - класен ръководител. Всеки предмет се води от точно един преподавател Course teacher (teacher). Всеки предмет е воден на точно един клас Course class (class_id), правим разлики дори в предмети с едно и също име, водени от един и същ преподавател. Всеки елемент на учебната програма представлява предмет воден в определен регулярен момент - Program course (course). Всеки изпит е по точно един предмет, но много изпити във времето може да са свързани с един предмет - Exam course (course). С всеки изпит са асоциирани един ученик и един предмет - Exam student (student) и Result exam (exam). Всеки ученик се асоциира с класовете, в които е бил докато е следва/л - Student in class (class_id). Връзката е много към много. С всеки ученик са свързани майка (може и да няма) - mother (mother), баща (може и да няма) - father (father) и основен родител за контакт main (main_parent).



3 E/R диаграма на модела на Базата данни



4 Преобразуване на Е/Р диаграмата до релационни схеми

При преобразуването са спазени следните принципи: на същност се съпоставя релация, на връзка от тип много към много се съпоставя релация с ключ първичните ключове на свързаните същности, за връзка от тип много към един съответно един към много се добавя атрибут към същността от страната на 'много', взето е предвид липсата на връзки от тип един към един и липсата на атрибути на свързките.

Teacher(id, first_name, last_name, gender, work_phone, email, address)

Class(id, lead_teacher, school_year,

number, letter, class_room, class_hour)

Course(id, teacher, class_id, name, semester)

Program_entry(id, course, day_of_week, room, starts, hours)

Exam(id, course, room, starts, hours)

Parent(id, first_name, last_name, gender,
phone, email, birthdate, address, work)

Student(id, mother, father, main_parent,

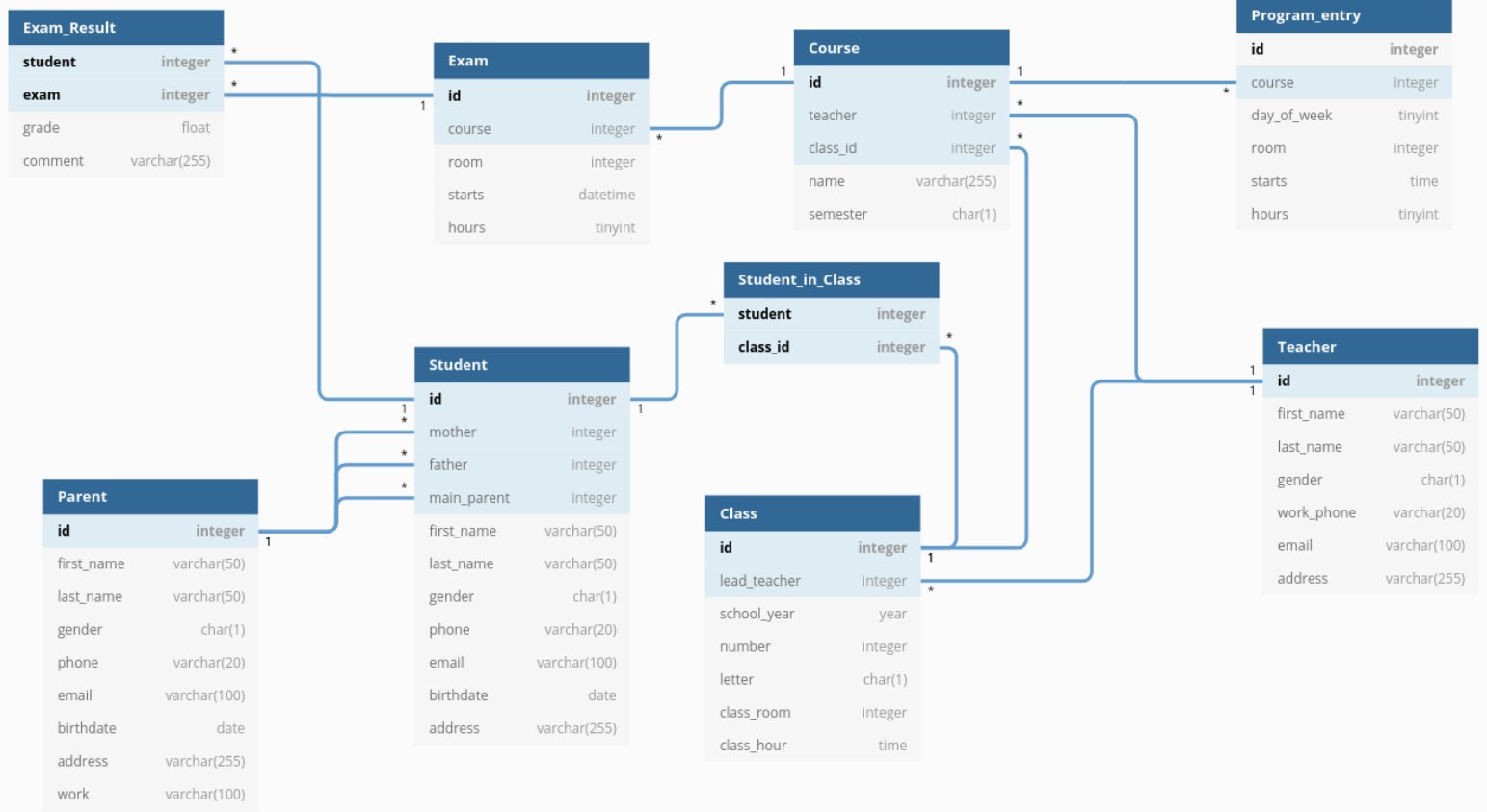
first_name, last_name, gender, phone, email, address)

Student_in_class(student, class_id)

Exam_Result(student, exam, grade, comment)

4.1 Ограничения по референтна цялост

$$\begin{aligned}\pi_{lead_teacher}(Class) &\subseteq \pi_{id}(Teacher) \\ \pi_{teacher}(Course) &\subseteq \pi_{id}(Teacher) \\ \pi_{class_id}(Course) &\subseteq \pi_{id}(Class) \\ \pi_{course}(Program_entry) &\subseteq \pi_{id}(Course) \\ \pi_{course}(Exam) &\subseteq \pi_{id}(Course) \\ \pi_{mother}(Student) &\subseteq \pi_{id}(Parent) \\ \pi_{father}(Student) &\subseteq \pi_{id}(Parent) \\ \pi_{main_parent}(Student) &\subseteq \pi_{id}(Parent) \\ \pi_{student}(Student_in_Class) &\subseteq \pi_{id}(Student) \\ \pi_{class_id}(Student_in_Class) &\subseteq \pi_{id}(Class) \\ \pi_{student}(Exam_Result) &\subseteq \pi_{id}(Student) \\ \pi_{exam}(Exam_Result) &\subseteq \pi_{id}(Exam)\end{aligned}$$



5 Описание на функциите

5.1 full_gender

Функцията full_gender по символ за пол връща пълното наименование на пола, ако той е 'M' или 'F', иначе връща 'Other'.

5.2 full_semester

Функцията full_semester по символ за срок връща пълното наименование на срока, ако той е 'F' или 'S', иначе връща 'Unknown'.

5.3 day_of_week_as_string

Функцията day_of_week_as_string по число за ден от седмицата връща пълното наименование на деня, ако аргумента е валиден ден от седмицата (цяло число между 1 и 5 включително), иначе връща 'Unknown'.

6 Описание на тригерите

6.1 Before_Insert_Exam_Result

Тригера се грижи за налагане на ограничение върху стойността на колоната grade - да бъде число в интервала [2.0, 3.0], задействано преди добавяне в таблицата Exam_Result.

6.2 Before_Update_Exam_Result

Тригера се грижи за налагане на ограничение върху стойността на колоната grade - да бъде число в интервала [2.0, 3.0], задействано преди обновяване в таблицата Exam_Result.

6.3 Before_Insert_Course

Тригера се грижи за налагане на ограничение върху стойността на колоната semester - да бъде или 'F' или 'S', задействано преди добавяне в таблицата Course.

6.4 Before_Update_Course

Тригера се грижи за налагане на ограничение върху стойността на колоната `semester` - да бъде или 'F' или 'S', задействано преди обновяване в таблицата `Course`.

6.5 Before_Insert_Program_entry

Тригера се грижи за налагане на ограничение върху стойността на колоната `day_of_week` - да бъде цяло число в списъка (1, 2, 3, 4 5), задействано преди добавяне в таблицата `Program_entry`.

6.6 Before_Update_Program_entry

Тригера се грижи за налагане на ограничение върху стойността на колоната `day_of_week` - да бъде цяло число в списъка (1, 2, 3, 4 5), задействано преди обновяване в таблицата `Program_entry`.

7 Описание на изгледите

7.1 Grade

Изгледа `Grade` предоставя пълна информация за една оценка. Информациите се извлича от таблиците: `Exam_Result`, `Student`, `Exam`, `Exam_Result`, `Course`, `Class` и `Teacher`. Като колоните, до които дава достъп изгледа са: `Exam_Result.student`, `Student.first_name`, `Student.last_name`, `Class.id`, `Class.school_year`, `Class.number`, `Class.letter`, `Course.id`, `Course.name`, `Course.semester`, `Teacher.id`, `Teacher.first_name`, `Teacher.last_name`, `Exam.starts`, `Exam_Result.grade` и `Exam_Result.comment`.

7.2 Program

Изгледа `Program` предоставя пълна информация за един учебен час от програмата. Информациите се извлича от таблиците: `Program_entry`, `Course`, `Class` и `Teacher`. Като колоните, до които дава достъп изгледа са: `Program_entry.id`, `Program_entry.day_of_week`, `Program_entry.room`, `Program_entry.starts`, `Program_entry.hours`, `Program_entry.course`, `Course.name`, `Course.semester`, `Course.teacher`, `Teacher.first_name`, `Teacher.last_name`, `Course.class_id`, `Class.school_year`, `Class.number`, `Class.letter`.

```
OkPacket {  
  fieldCount: 0,  
  affectedRows: 0,  
  insertId: 0,  
  serverStatus: 10,  
  warningCount: 0,  
  message: '',  
  protocol41: true,  
  changedRows: 0  
}
```

8 Описание на приложението за достъп с резултатите от заявката

Приложението за достъп е изключително просто (семпло). Написано е на езика JavaScript и ползва пакета `mssql` за свързване с `mysql` сървър. Приложението се свързва към сървъра посредством забита в кода информация за потребител, парола и база, които всички са `test-drive`. След, което чете подадените чрез командните параметри файл, който трябва да е валиден `sql` скрипт, не съдържащ командни `DELIMITER` понеже, те не се поддържат от използвания пакет. След свързване, то изпълнява заявките от скрипта и извежда резултат.

8.1 Резултати при изпълнение с `combined_sql/test.sql`

Скрипта създава таблиците, попълва ги, след което създава изгледите и накрая прави заявка към всеки изглед. Резултатите се подобни на тези, ако използване CLI-то на `mysql`.

8.1.1 Създаване на таблица

8.1.2 Вмъкване в таблица

8.1.3 Ред, резултат от заявка към `Grade`

8.1.4 Ред, резултат от заявка към `Program`

```
OkPacket {
  fieldCount: 0,
  affectedRows: 3,
  insertId: 0,
  serverStatus: 10,
  warningCount: 0,
  message: '&Records: 3 Duplicates: 0 Warnings: 0',
  protocol41: true,
  changedRows: 0
}
```

```
RowDataPacket {
  student: 3,
  student_first_name: 'Jordan',
  student_last_name: 'Angelov',
  class: 1,
  school_year: 2016,
  class_number: 12,
  class_letter: 'B',
  course: 1,
  course_name: 'Mathematics',
  course_semester: 'F',
  teacher: 1,
  teacher_first_name: 'Veselka',
  teacher_last_name: 'Hristova',
  exam_date: 2016-01-15T08:00:00.000Z,
  grade: 6,
  comment: null
}
```



```
RowDataPacket {  
  entry: 3,  
  day_of_week: 3,  
  room: 32,  
  starts: '11:20:00',  
  hours: 3,  
  course: 5,  
  subject: 'Technology of Programming',  
  semester: 'F',  
  teacher: 2,  
  teacher_first_name: 'Kiril',  
  teacher_last_name: 'Mitov',  
  class_id: 1,  
  school_year: 2016,  
  class_number: 12,  
  class_letter: 'B'  
}
```