

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Астраханский государственный университет» (АГУ)

Допускается к защите

«_____» _____ 20__ г.

Завкафедрой _____

ДИПЛОМНАЯ РАБОТА

**«БАЗА ДАННЫХ „ПОДГОТОВКА КАДРОВ И
ПРОФЕССИОНАЛЬНАЯ КВАЛИФИКАЦИЯ“
ДЛЯ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ
СИСТЕМЫ „НАУЧНАЯ ДЕЯТЕЛЬНОСТЬ“»**

Выполнила

студентка группы ПМ-51

Данилова Т.С.

Научный руководитель

к.т.н., доцент кафедры ПМИ

Зелепухина В.А.

Астрахань – 2013

Содержание

ВВЕДЕНИЕ.....	3
Глава 1. ЛОГИКО-КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1. Основные действующие лица предметной области.....	5
1.2. Структура базы данных.....	10
Глава 2. ОПИСАНИЕ ПОДСИСТЕМЫ «ПОДГОТОВКА КАДРОВ И ПРОФЕССИОНАЛЬНАЯ КВАЛИФИКАЦИЯ».....	21
2.1. Программа для добавления/редактирования сведений	21
2.2. Программа для фильтрации/поиска информации об обучении/защитах	30
2.3. Рейтинг диссертаций	39
ЗАКЛЮЧЕНИЕ	42
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	43
ПРИЛОЖЕНИЯ.....	47

ВВЕДЕНИЕ

Актуальность. В настоящее время актуальна проблема мониторинга научной активности отдельных сотрудников и подразделений в целом. Системы учета научной активности разрабатываются как в нашей стране (ИСТИНА МГУ [1], ИАС НИД СПбГУ [2], БелГУ [3]), так и за рубежом (Edinburgh Research Explorer [4]). Подобные системы позволяют автоматизировать процессы сбора, анализа и представления в удобной форме информации о научной деятельности организации, её подразделений.

В Астраханском государственном университете реализуется проект «Разработка системы сбора, структурирования, анализа и представления научной и наукометрической информации на уровне научной организации (подразделения)» [5, 6]. Система представляет собой корпоративное веб-приложение, позволяющее пользователям системы формировать информацию о результатах научной деятельности (РНД): публикации, интеллектуальная собственность, научные проекты и их финансирование, сведения о признании.

Одним из показателей для оценки качества работы учёного и подразделения, в котором он работает, является руководство научно-исследовательской работой аспирантов и докторантов. Кроме того, информация о профессиональной квалификации учитывается при подготовке различных отчетов.

В связи с этим в рамках проекта необходима подсистема, включающая базу данных (БД) «Подготовка кадров и профессиональная квалификация» и программный модуль, позволяющий пользователю (сотруднику организации) наполнять её информацией по двум направлениям: обучение в аспирантуре/докторантуре, защиты диссертаций, при этом:

1. Указанная подсистема должна вписываться в общую концепцию информационно-аналитической системы «Научная деятельность» [7], допол-

нять её новыми функциональными возможностями, не нарушая при этом ядро программного комплекса.

2. Одна запись об обучении/защите может быть связана с несколькими сотрудниками. Например, единожды введенная информация об обучении в аспирантуре автоматически переносится на персональные страницы «ученика» и научных руководителей. Таким образом, устраняется избыточность базы данных и уменьшается время, необходимое для ее наполнения.

Целью дипломной работы является автоматизация сбора, хранения и представления в удобной форме сведений о профессиональной квалификации сотрудников организации.

Для достижения цели были поставлены следующие **задачи**:

1. Проведение анализа предметной области и ознакомление с информационно-аналитической системой «Научная деятельность».

2. Разработка концепции и структуры подсистемы «Подготовка кадров и профессиональная квалификация».

3. Разработка структуры БД для накопления информации, необходимой подсистеме.

4. Разработка, отладка, внедрение и адаптация подсистемы для сбора информации об обучении/защитах и её фильтрации/поиска.

5. Автоматизация ввода данных об обучении/защитах путём загрузки файла автореферата диссертации, анализа информации, представленной в нем и идентификации связей с сотрудниками организации.

Исследования проведены при финансовой поддержке Российского гуманитарного научного фонда (грант № 12-03-12000) и Российского фонда фундаментальных исследований (грант № 12-07-31145).

Глава 1. ЛОГИКО-КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ

Система показателей в баллах для аттестации и определения квалификационных категорий профессорско-преподавательского состава (ППС) включает показатели [8]:

- Научное руководство подготовкой кандидатской диссертации, научное консультирование при подготовке докторской диссертации.
- Количество кандидатских (докторских) диссертаций, выполненных под руководством (консультированием) и защищенных в срок.

Кроме того, для определения значений показателей сотрудников/подразделений важна информация не только о факте присуждения ученой степени кандидата/доктора наук, но и сведения об оппонировании диссертаций, о количестве защит «в срок», о количестве аспирантов, выбывших из аспирантуры до окончания срока обучения.

Сложность разработки модели и структуры хранения информации о профессиональной квалификации связана с тем, что ни один из существующих форматов метаданных, таких как Current European Research Information Format (CERIF) [9], не отражает подобные данные в полной мере. Важна не только связь сотрудника с конкретным обучением или защитой, но и его роль в нем — ученик, научный руководитель, оппонент: совокупность таких атрибутов, в целом, влияют на персональный рейтинг сотрудника и рейтинг подразделения, в котором он зарегистрирован.

1.1. Основные действующие лица предметной области

Для правильного анализа функциональной деятельности организации, а также для последующего выделения объектов и описания их атрибутов, необходимо детально изучить процесс подготовки научных и научно-педагогических кадров высокой квалификации.

Ученая степень кандидата или доктора наук присуждается лицам, получившим результаты в научно-исследовательской работе. Несмотря на кажущуюся однозначную связь между обучением и фактом защиты, необходимо разделять эти процессы, потому как защите кандидатских/докторских диссертаций не всегда предшествует обучение соискателя в аспирантуре/докторантуре, соответственно.

Подготовка диссертации на соискание ученых степеней осуществляется двумя основными способами:

1. в процессе обучения в аспирантуре (докторантуре) вуза или научной организации;
2. путём прикрепления соискателем к вузу или научной организации для подготовки диссертации.

Иными словами, во втором случае соискатель подготавливает диссертацию на соискание ученой степени кандидата (доктора) наук без обучения в аспирантуре (докторантуре).

Кроме того, статус обучающегося в случае отчисления из аспирантуры/докторантуры может характеризоваться следующими параметрами: отчислен, отчислен с правом восстановления, отчислен с представлением диссертации на кафедру, отчислен с представлением диссертации в совет, отчислен с защитой (Рис. 1).

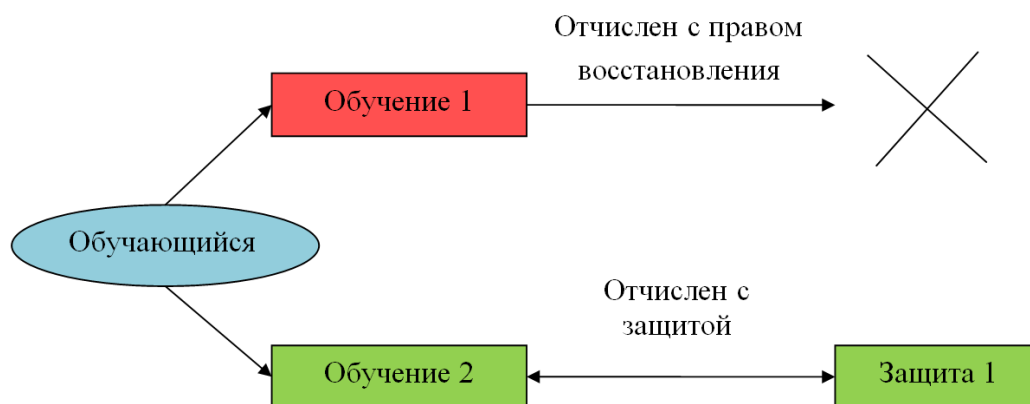


Рис. 1. Статусы обучающегося в случае отчисления

Таким образом, мы не можем реализовать в БД отношение со следующей структурой (Таблица 1).

Таблица 1

Неверная структура БД

<u>код записи</u>	код сотрудника	год начала обучения	год окончания обучения	название диссертации	дата защиты	номер дис. совета	...
-----------------------	----------------	---------------------	------------------------	----------------------	-------------	-------------------	-----

Данному отношению свойственна избыточность данных, и могут возникать так называемые аномалии обновления записей [10]. Если была внесена некорректная информация об обучении или же её вовсе стоит удалить из базы данных, то и соответствующие атрибуты, отражающие сведения о защите, также будут удалены, что противоречит правилам нормализации реляционной модели данных.

Определим основных действующих лиц рассматриваемого процесса:

1. **Участник** — лицо, принимающее участие в обучении или защите.
2. Ученик, научные руководители — **участники обучения**.
3. Соискатель, научные руководители, оппоненты — **участники защит**.
4. **Подразделение** — структура, являющаяся частью организации, для которой разрабатывается и внедряется информационная система.
5. **Сотрудник** — лицо, работающее в подразделении.
6. **Персона** — лицо, не являющееся сотрудником организации, либо сотрудник, не зарегистрированный в информационной системе. Предполагается, что в дальнейшем сразу после регистрации для пользователя будут определяться те записи об обучении/защите, в которых участником является персона, данные которой похожи на данные текущего пользователя:

в таком случае пользователю останется только подтвердить свое отношение в той или иной записи.

Диаграмма связи между действующими лицами системы построена с использованием нотации UML [11] (Рис. 2).

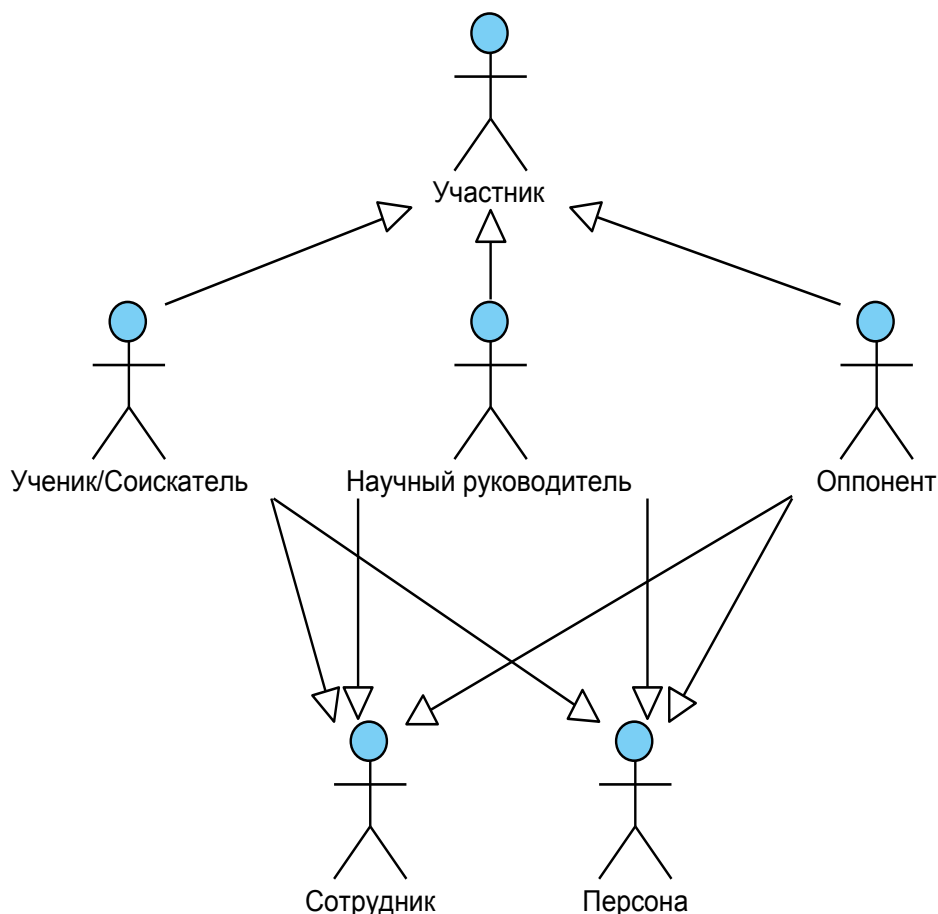


Рис. 2. Связи между действующими лицами системы

Основные **принципы** разрабатываемой подсистемы (Рис. 3):

1. Сведения об обучении/защите добавляются в БД только в том случае, если участниками процесса являются **сотрудники организации**, поэтому необходимо определять связь информации о процессе с сотрудником организации.

2. В том случае, если сотрудник не зарегистрирован в информационной системе, связь определяется через **подразделение**, в котором работает (работал) сотрудник на момент обучения/защиты.

3. Информация об обучении/защите может иметь **множественные связи с подразделениями**. Например, научный руководитель работает на кафедре прикладной математики и информатики, а его аспирант — на кафедре информационных систем. В данном случае информация должна учитываться в работе обеих кафедр.

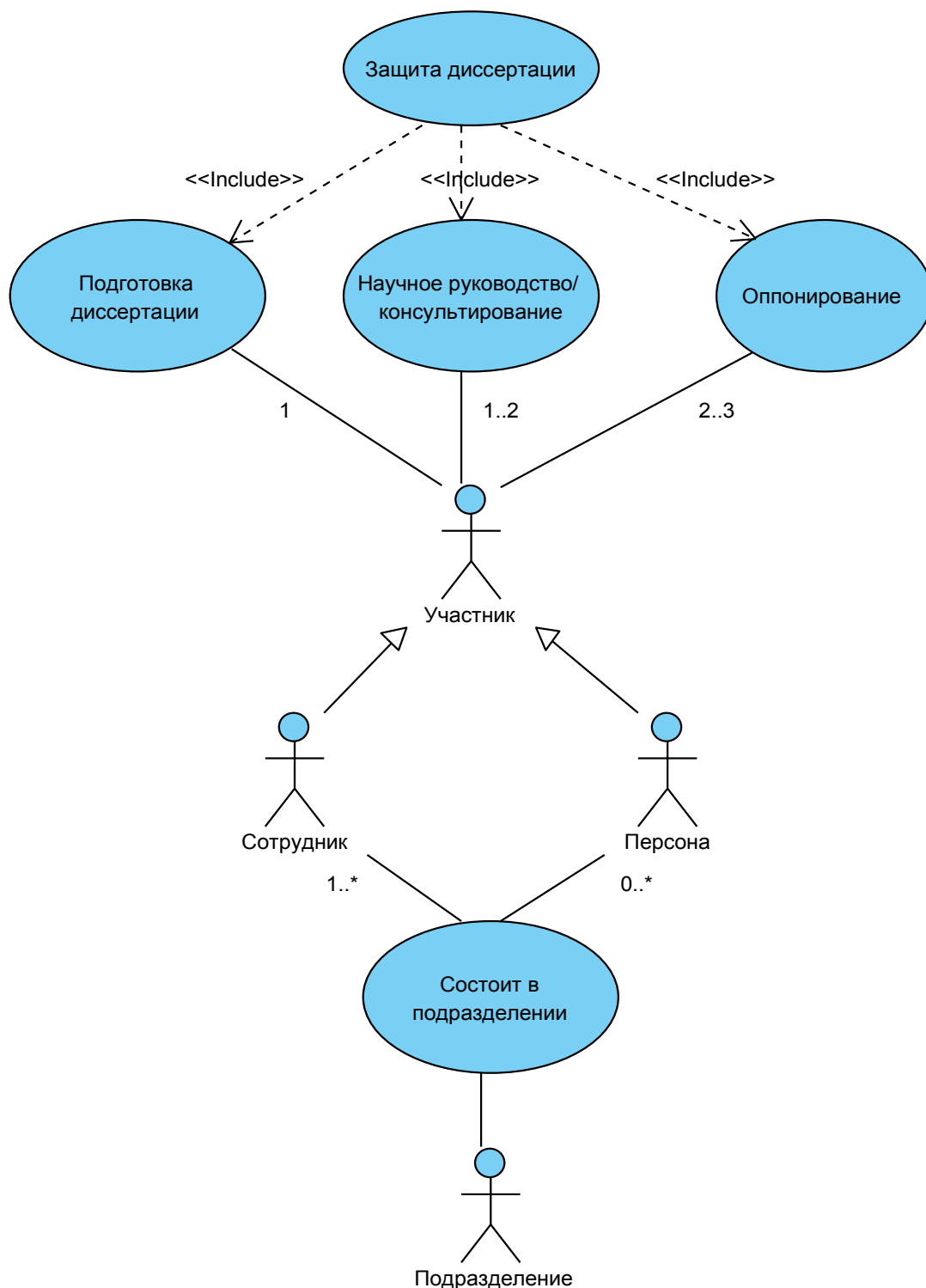


Рис. 3. Функциональные связи между действующими лицами

Согласно рассмотренным принципам, мы определили **требования**, предъявляемые к подсистеме учета кадров высшей квалификации:

1. Обучение в аспирантуре/докторантуре не обязательно заканчивается защитой и наоборот (обучение и защита — два разных объекта).
2. В БД могут храниться записи об обучении или защите человека, не являющегося на данный момент сотрудником (бывший сотрудник).
3. В каждом обучении принимает участие ровно один ученик.
4. В каждой защите принимает участие ровно один соискатель.
5. В каждом обучении и защите указываются не более двух консультантов в случае докторской диссертации.
6. В каждом обучении и защите указываются не более двух научных руководителей для кандидатской диссертации.
7. В защите кандидатской диссертации принимают участие ровно два оппонента, в докторской — ровно три.
8. Имена участников процессов на момент обучения и защиты могут не совпадать с текущими (смена фамилии в силу ряда причин).
9. Для каждой записи об обучении/защите должна быть создана связь с сотрудником и (или) подразделением.
10. Защита диссертации может осуществляться по нескольким специальностям.

1.2. Структура базы данных

С учетом вышеперечисленных принципов и требований, был определен набор отношений БД, являющихся частью БД «Научная деятельность», в составе которой были реализованы некоторые действующие лица (**Сотрудники, Подразделения**).

Данные о сотрудниках отражены в таблице `default_users`. Эта таблица хранит информацию об имени пользователя, электронной почте, идентификаторе (первичном ключе) и т.д. По аналогии таблица

default_xlaps_departments хранит информацию о подразделениях: название, описание, идентификатор (первичный ключ) и т.д. Отношение СОТРУДНИК-ПОДРАЗДЕЛЕНИЕ характеризуется видом связи «многие-ко-многим» (сотрудник может принадлежать нескольким подразделениям, в свою очередь в подразделении числятся несколько сотрудников). Данное отношение реализовано в БД «Научная деятельность» посредством таблицы default_xlaps_departments_workers. За реализацию связи СОТРУДНИК-ПОДРАЗДЕЛЕНИЕ отвечают внешние ключи таблиц default_users, default_xlaps_departments — user_id и department_id, соответственно.

В БД «Подготовка кадров и профессиональная квалификация» определены новые отношения, описывающие следующие объекты: **Персоны, Обучения, Защиты.**

Данные о **Персонах** (лицах, не являющихся сотрудниками) хранятся в таблице default_xlaps_person. Атрибуты данного отношения включают в себя: идентификатор (id — первичный ключ), имя (full_name), степень (degree), звание (title).

Отношение default_xlaps_education хранит основную информацию об обучении в аспирантуре/докторантуре: тип обучения (education_type_id), год начала (start_year), статус начала (start_status_id), год окончания (ending_year), статус окончания (ending_status_id), специальность (specialty_id), организация (organization_id), город (city_id) (Рис. 4).

Для устранения избыточности информации, в частности, во избежание повторного ввода информации о городах, организациях и т.д., структура БД была подвергнута нормализации, для чего введены дополнительные отношения (*Таблица 2*).

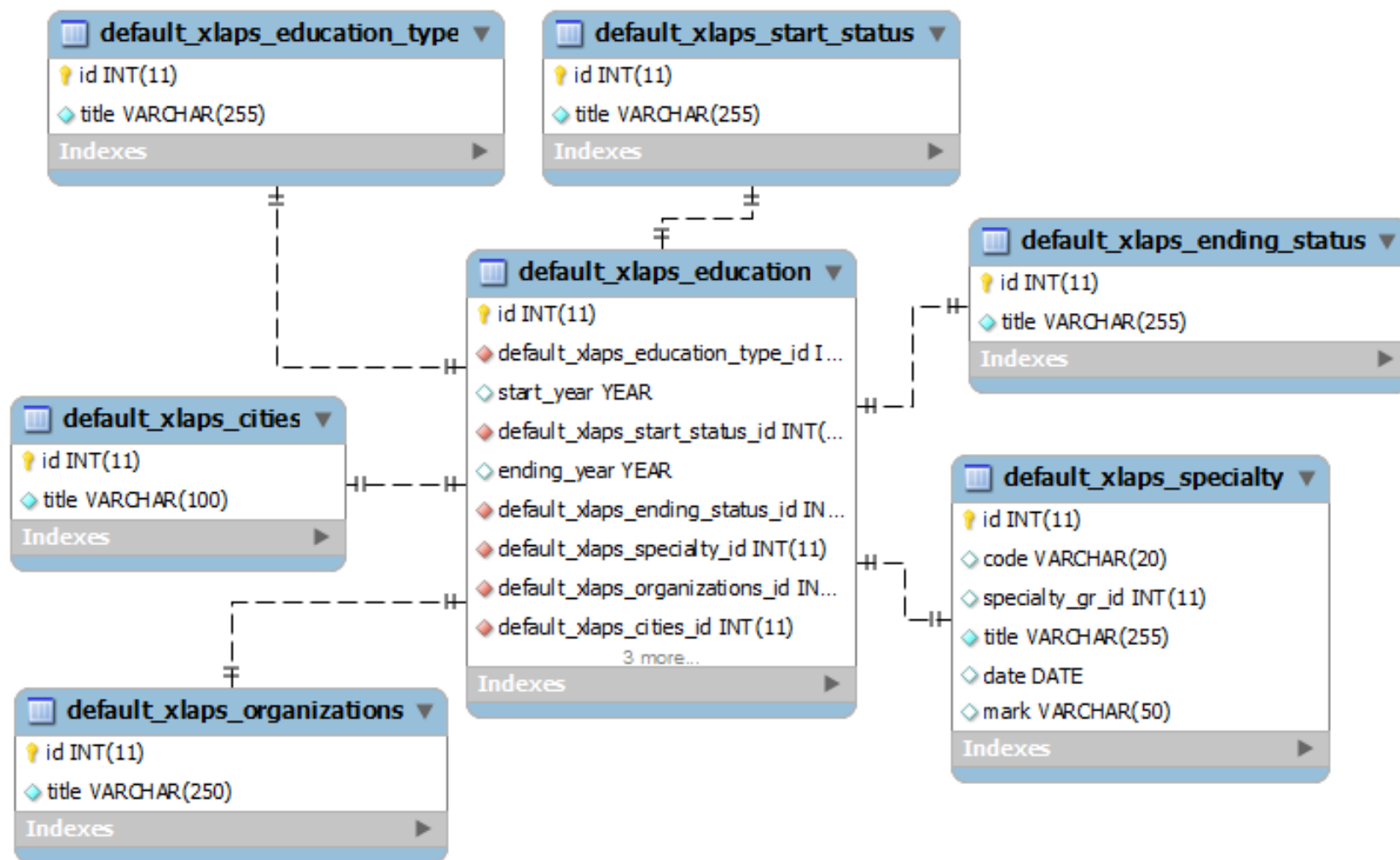


Рис. 4. ER-диаграмма

Информация о внешних ключах отношений

Внешний ключ	Отношение	Краткое описание
education_type_id	default_xlaps_education_type	типы обучения (аспирантура, докторантура)
start_status_id	default_xlaps_start_status	статусы начала обучения (поступил, восстановился)
ending_status_id	default_xlaps_ending_status	статусы окончания обучения
specialty_id	default_xlaps_specialty	список специальностей
organization_id	default_xlaps_organization	список организаций
city_id	default_xlaps_city	список городов

Отношение default_xlaps_defense хранит основную информацию о защитах кандидатских и докторских диссертаций: тип защиты (defense_type_id), дата защиты (defense_date), статус защиты (defense_status_id), дата утверждения (approval_date), название диссертации (title), отрасль наук (branch_science_id), организация (organization_id), город (city_id), номер диссертационного совета (dissertation_council_number_id), ведущая организация (lead_organization), ссылка на автореферат (file_url) (Рис. 5).

В связи с тем, что специальностей у некоторых диссертаций может быть несколько, в структуру добавлено отношение default_xlaps_defense_specialty, где defense_id и specialty_id — внешние ключи для таблиц default_xlaps_defense и default_xlaps_specialty, соответственно (Рис. 6).

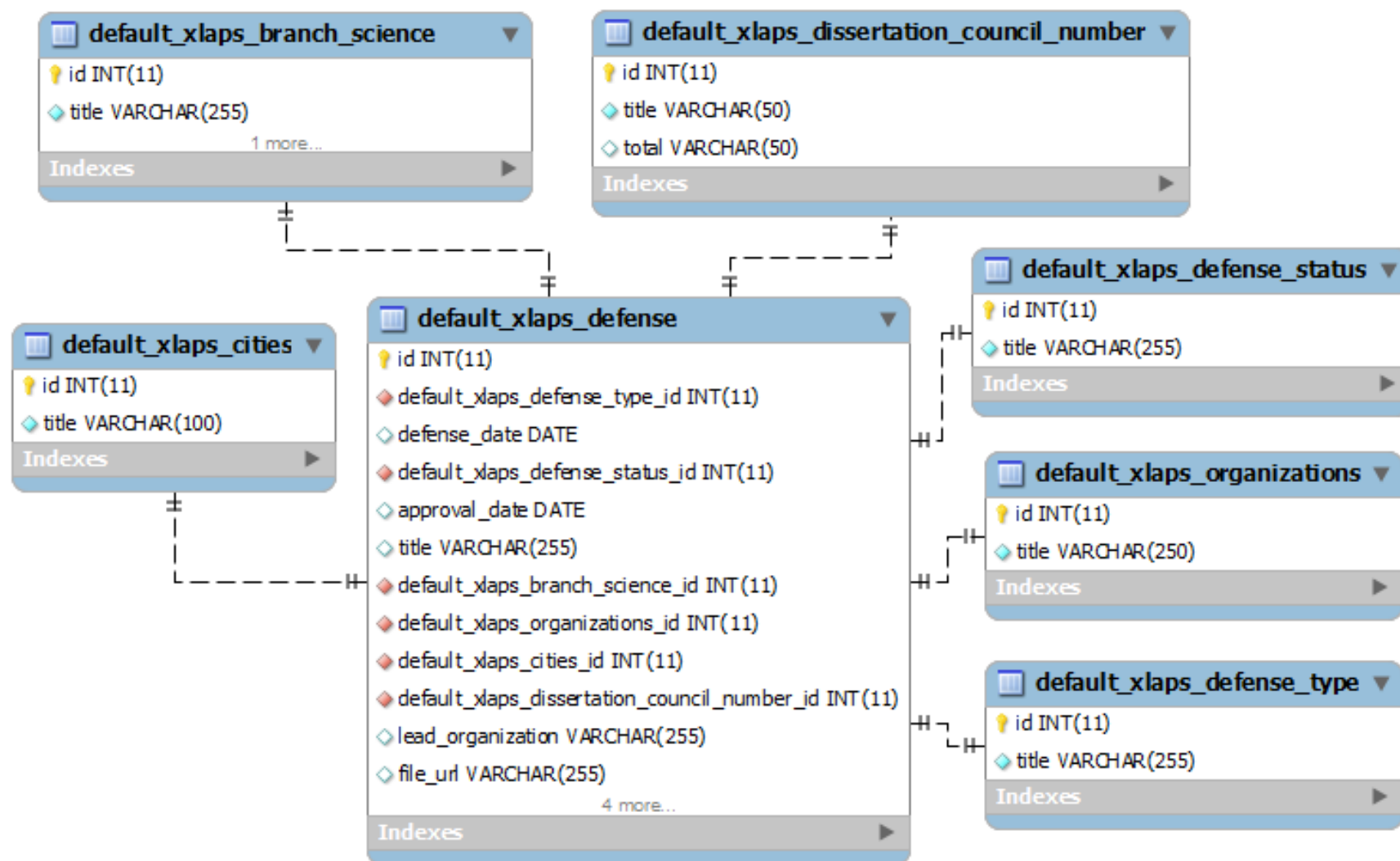


Рис. 5. ER-диаграмма

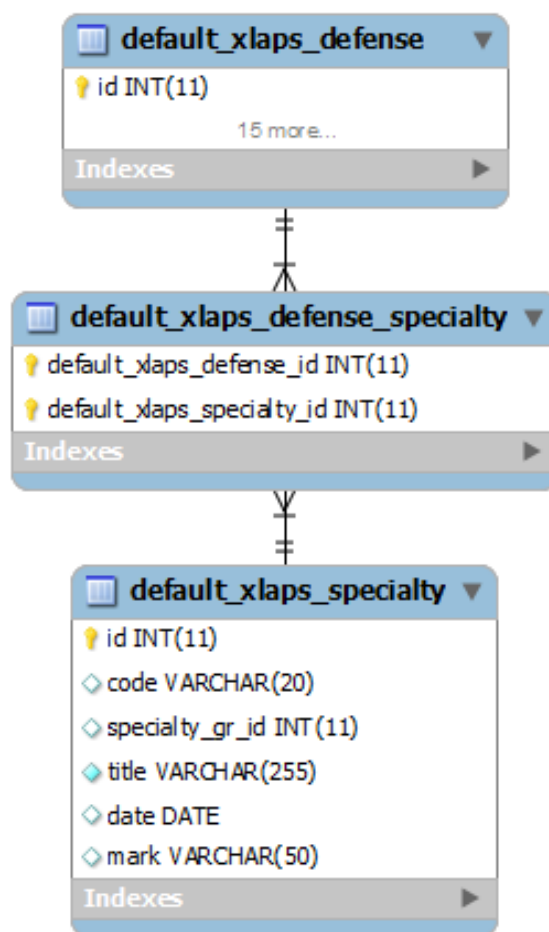


Рис. 6. ER-диаграмма

Действующая номенклатура специальностей научных работников взята с сайта Высшей аттестационной комиссии (ВАК) [12]. Шифр и название специальностей хранятся в атрибутах `code` и `title` соответственно.

Номенклатура специальностей научных работников периодически обновляется, поэтому может возникнуть ситуация, когда пользователь системы будет добавлять сведения о защите диссертации, информация о специальности которой отсутствует в действующей номенклатуре. Поэтому в отношение были добавлены два дополнительных атрибута: `date` (дата обновления) и `mark` (для определения источника обновления: действующий список ВАК или пользователь информационной системы).

Кроме того, для действующего списка специальностей ВАК определена связь с отношением «Отрасли науки» (Рис. 7).

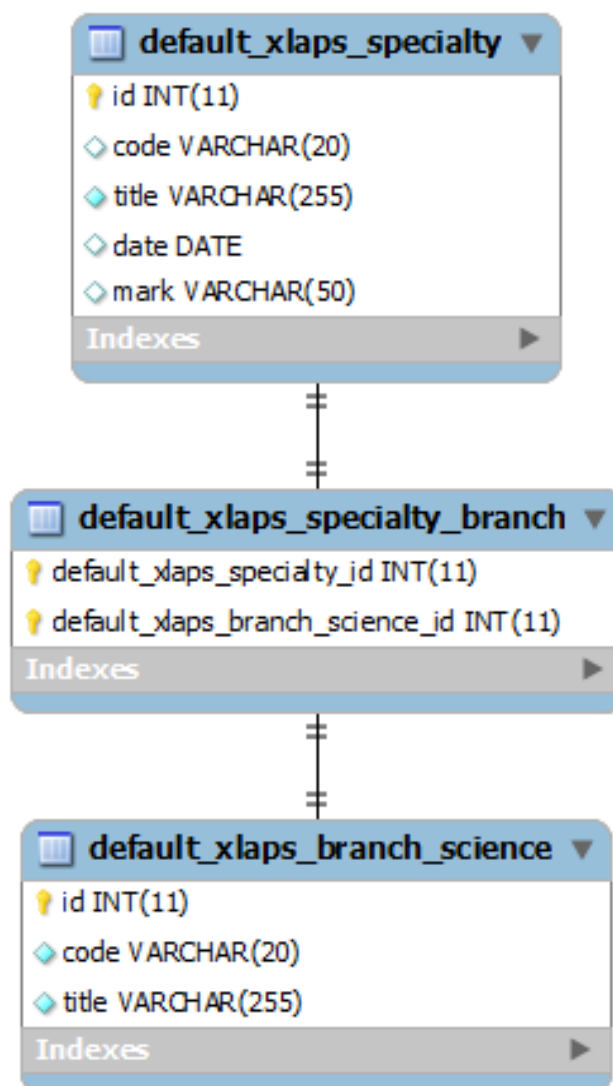


Рис. 7. ER-диаграмма

Связь между обучением и защитой осуществляет отношение `default_xlaps_education_defense`, содержит атрибуты `education_id`, `defense_id` — внешние ключи таблиц `default_xlaps_education` и `default_xlaps_defense` соответственно (Рис. 8).

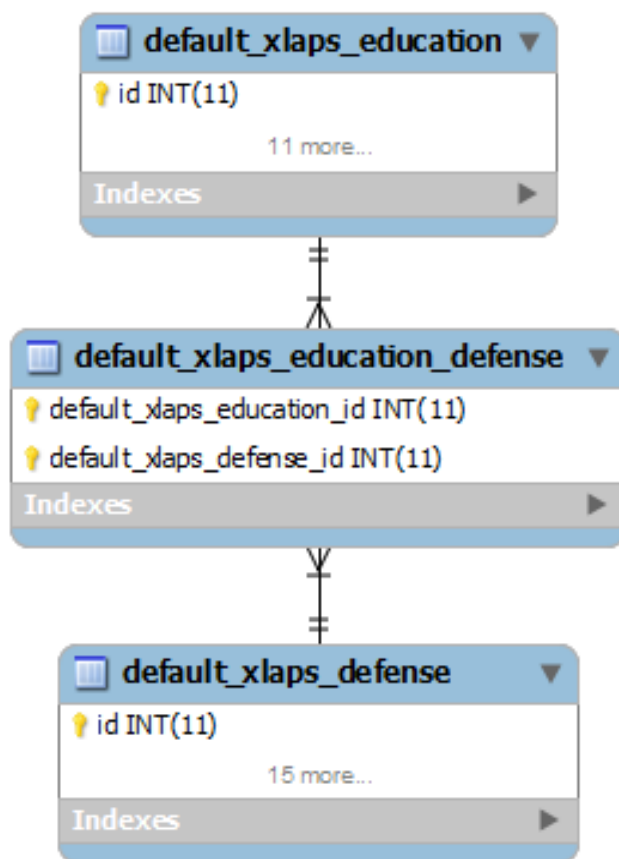


Рис. 8. ER-диаграмма

За связь соискателя с защитой отвечают поля `student_id` (код студента) и `student_table` (таблица-источник информации о студенте). Поле `student_id` является внешним ключом по отношению к таблицам `default_users` и `default_xlaps_person`. Поле `student_table` принимает значение «users» или «person» в зависимости от того, внешним ключом какой таблицы является поле `student_id` (Таблица 3). Поле `student_name` хранит имя соискателя на момент защиты.

Таблица 3

Пример заполнения отношения `default_xlaps_defense`

<u>id</u>	student_id	student_table	student_name
1	1	users	Ю.Ю. Тарасевич
2	1	person	И.И. Иванов

В защите могут принимать участие несколько научных руководителей и оппонентов, такой вид отношения называется «многие-ко-многим». Для реализации данного типа отношений вводится таблица `default_xlaps_defense_members`, осуществляющая связь данных таблиц `default_xlaps_defense` и `default_users`, `default_xlaps_person` (Таблица 4).

Таблица 4

Пример заполнения таблицы `default_xlaps_defense_members`

defense_id	member_id	member_table	member_name	member_role
1	1	users	Ю.Ю. Тарасевич	supervisor
2	1	person	И.И. Иванов	opponent

Поле `defense_id` является внешним ключом по отношению к таблице `default_xlaps_defense`, `member_id` — внешний ключ по отношениям к таблицам `default_users` и `default_xlaps_person`. Поле `member_table` принимает значение «users» или «person» в зависимости от того внешним ключом какой таблицы является поле `member_id`. Поле `member_role` указывает на роль участника в защите: «supervisor» — научный руководитель, «opponent» — оппонент.

За связь **Персоны с Подразделениями** отвечает отношение типа «многие-ко-многим» — `default_xlaps_defense_member_department`. Атрибутами отношения являются: `defense_id` (код защиты), `member_id` (код участника), `member_table` (таблица участника), `member_role` (роль участника), `department_id` (код подразделения).

К отношениям типа «многие-ко-многим» так же относятся связи:

1. «Участник защиты — степень на момент защиты» (таблица `default_xlaps_defense_member_degree`).
2. «Участник защиты — звание на момент защиты» (таблица `default_xlaps_defense_member_title`).

Структура связей между **Защитами** и **Сотрудниками**, **Персонами**, **Подразделениями** изображена на диаграмме (Рис. 9).

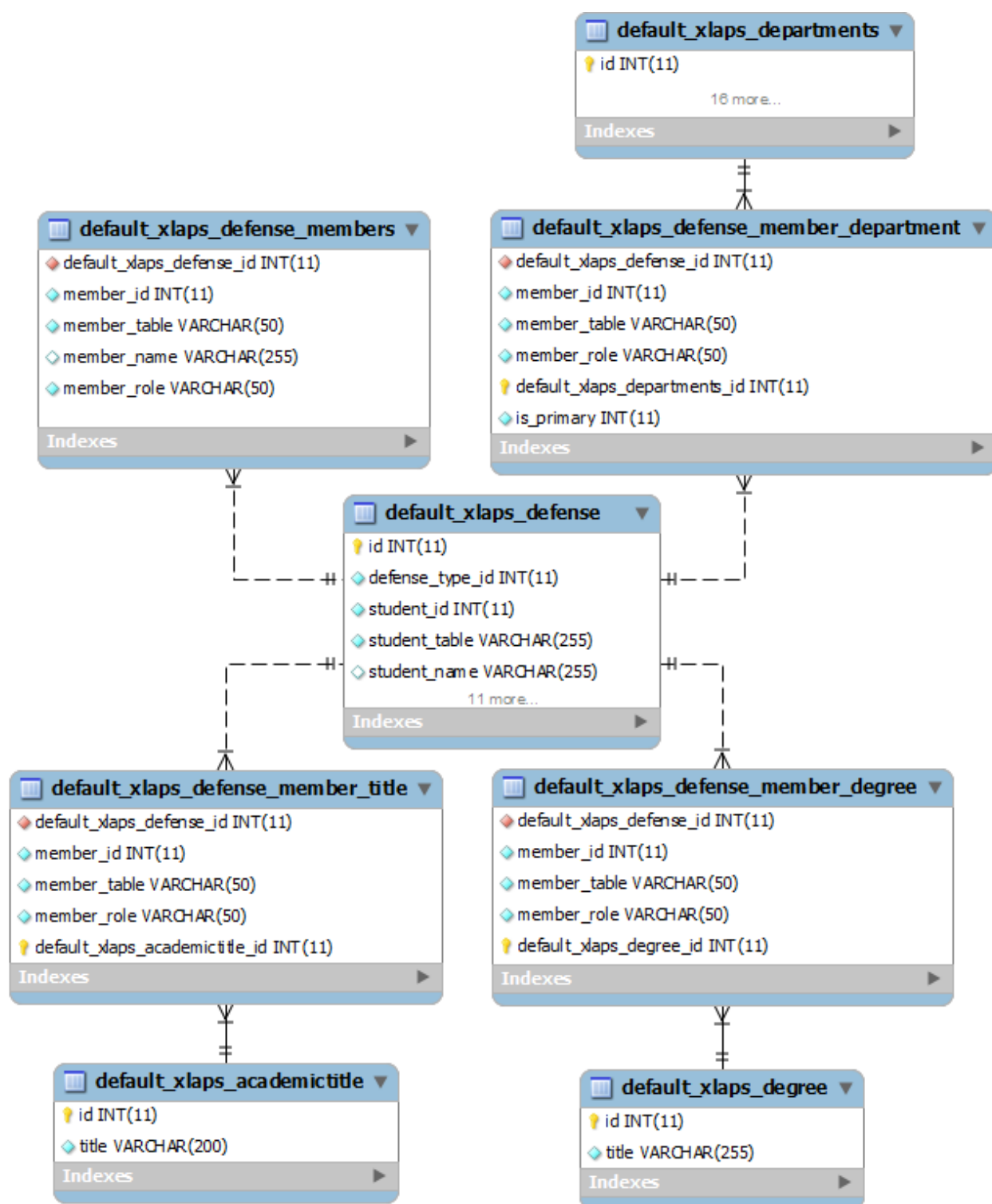


Рис. 9. ER-диаграмма

Связи обучения с сотрудниками, подразделениями и персонами строятся аналогично (Рис. 10).

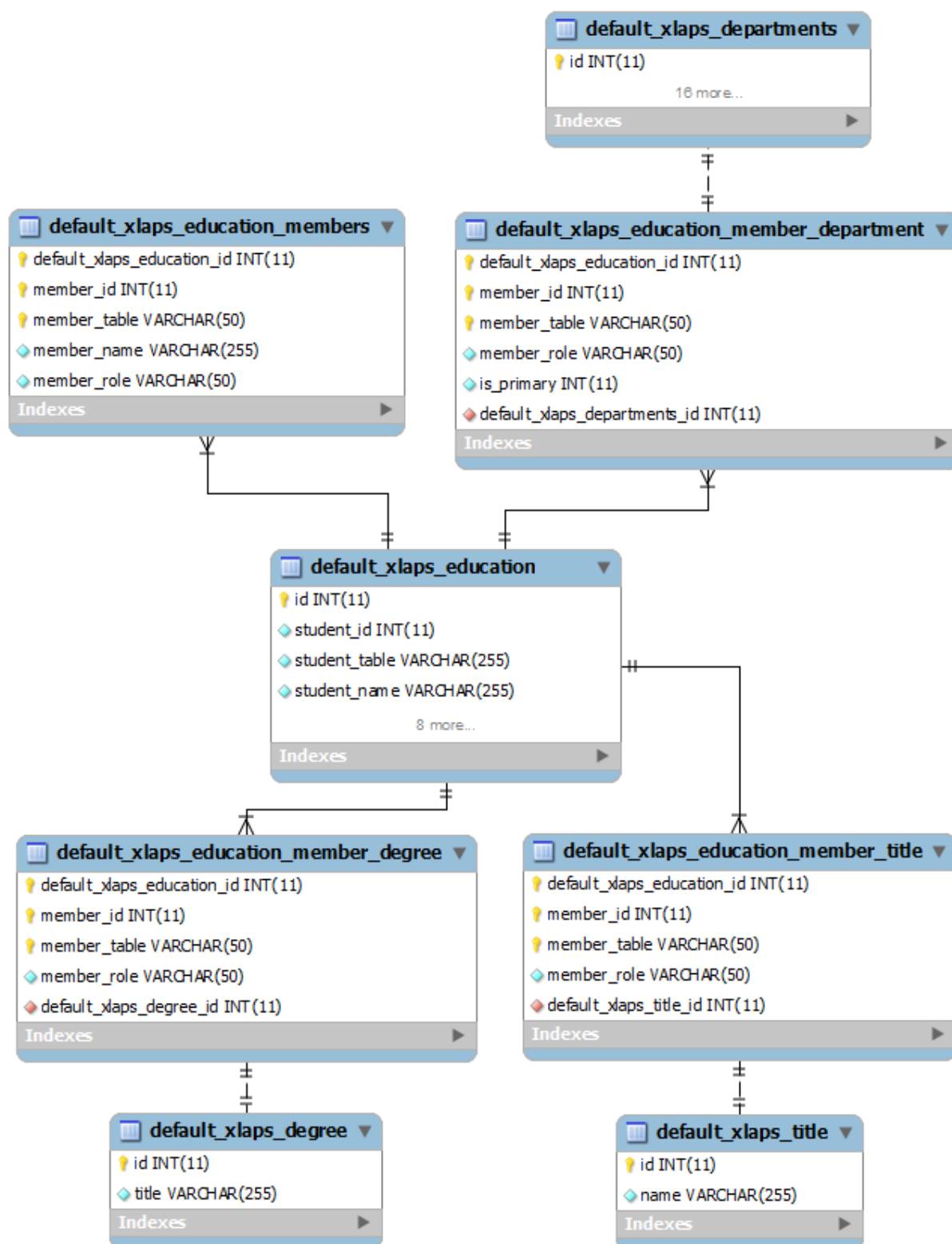


Рис. 10. ER-диаграмма

Глава 2. ОПИСАНИЕ ПОДСИСТЕМЫ «ПОДГОТОВКА КАДРОВ И ПРОФЕССИОНАЛЬНАЯ КВАЛИФИКАЦИЯ»

Информационно-аналитическая система «Научная деятельность» разрабатывается с использованием языка программирования PHP5 [13], СУБД MySQL 5.1 [14]. Подпрограммы для добавления/редактирования сведений о публикациях, грантах, патентах и т.д. строятся на основе CMF (Content Management Framework) CodeIgniter 2.0 [15] и библиотеки grocery CRUD 1.3.3 [16]. С целью интеграции в главное приложение и совместимости с остальными модулями, в процессе разработки подсистемы «Подготовка кадров и профессиональная квалификация» был использован тот же набор инструментария.

2.1. Программа для добавления/редактирования сведений

Обрабатываемая программой информация сгруппирована в два блока:

1. информация об аспирантуре, докторантуре:
 - a. собственное обучение,
 - b. научное руководство (консультирование) учеников;
2. информация о защитах диссертаций:
 - a. собственные защиты (кандидатские, докторские),
 - b. защиты аспирантов и докторантов,
 - c. оппонирование кандидатских и докторских диссертаций.

В личном кабинете им соответствуют два раздела — Аспирантура/докторантура и Защиты (Рис. 11).

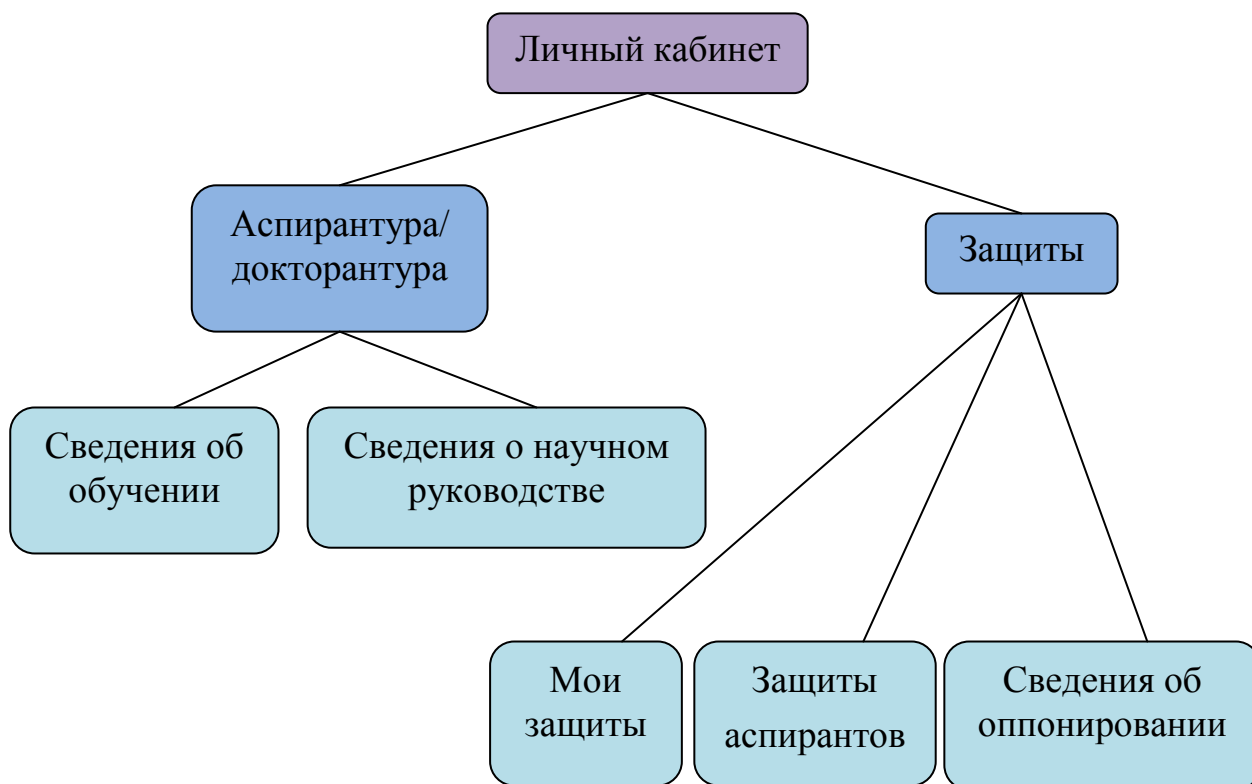


Рис. 11. Навигация по подсистеме

Для работы с информацией о защитах диссертаций создан класс `Defenses`. Глобальная переменная `$manage_type` служит для определения, по какому направлению будет заноситься информация о защите диссертации — собственные защиты, защиты аспирантов, сведения об оппонировании. Такое разделение необходимо для обеспечения контролируемого доступа к информации.

При выборе пользователем какого либо из подразделов раздела Защиты, происходит вызов одного из методов класса: `defense`, `guide`, `opposition`, в каждом из которых переопределяется значение переменной `$manage_type` и происходит вызов главного метода класса - `_manage`, реализующего CRUD-операции с таблицами БД (CRUD — Create Read Update Delete) посредством библиотеки `grocery CRUD`.

```
function defense ($operation = null) {  
    $this->manage_type = "defense";  
    $this->_manage($operation);  
}
```

После подключения библиотека автоматически генерирует веб-интерфейс для работы с данными указанной таблицы БД (Рис. 12), а также конструирует соответствующие SQL-запросы (для обновления, удаления, вставки, выборки).

Для автоматической генерации полей, которые будут отображаться на форме добавления и редактирования в соответствии с типом поля БД, необходимо перечислить их имена в качестве параметров функции `fields`:

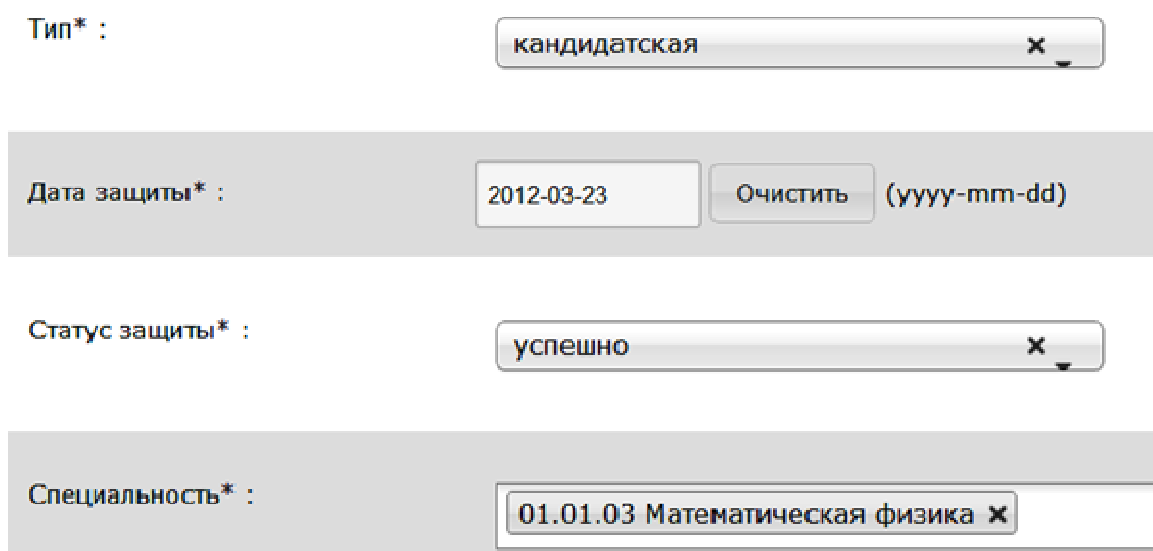
```
$crud->fields ('defense_type_id', 'defense_date',  
'defense_status_id', ...);
```

Функция `required_fields` определяет поля, обязательные для заполнения:

```
$crud->required_fields ('defense_type_id',  
'defense_date', ...);
```

Библиотека позволяет реализовывать зависимости типа «один-к-одному», «один-ко-многим» и «многие-ко-многим». Например, тип обучения («один-ко-многим»):

```
$crud->set_primary_key ('id', 'xlaps_defense_type');  
$crud->set_relation ('defense_type_id',  
'xlaps_defense_type', 'title');
```



Тип* :

Дата защиты* : (yyyy-mm-dd)

Статус защиты* :

Специальность* :

Рис. 12. Некоторые поля для ввода основной информации о защите

Сложность возникает при разработке интерфейса для добавления сведений о соискателе, научных руководителях и оппонентах. Структура БД такова, что каждый из участников защиты может являться как сотрудником, так и персоной. Определение такого типа отношений невозможно штатными методами библиотеки.

Рассмотрим форму для добавления научного руководителя (Рис. 13).

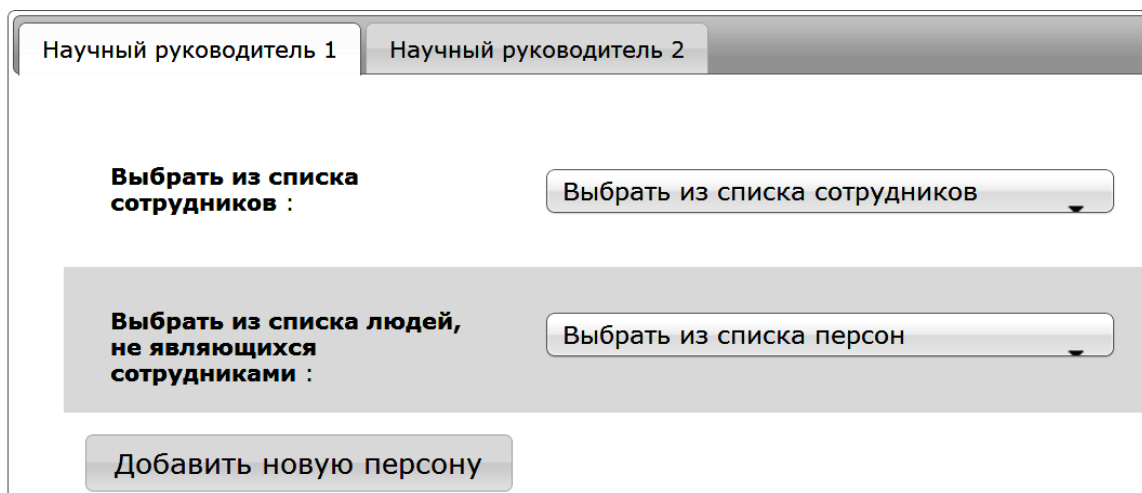


Рис. 13. Форма для ввода информации о научном руководителе

Двум спискам на форме (список сотрудников и список персон) соответствуют два поля `sv1_user_id` и `sv1_person_id`, не ассоциированных с таблицей БД. Типы этих полей были переопределены с помощью функций `_callback_field_sv1_users` и `_callback_field_sv1_person`, посредством которых были реализованы зависимости типа 1-n с таблицами `default_users` и `default_xlaps_person` в БД. Эти и аналогичные функции собраны в классе `Defence_Common`.

С использованием библиотеки jQuery [17] разработана коллекция функций для реализации пользовательского интерфейса.

Поля для ввода информации о научных руководителях и оппонентах разбиты на вкладки: научный руководитель 1, научный руководитель 2, оппонент 1 и т.д. За работу с каждой из таких вкладок отвечает функция `supervisors_opponent`, которая позволяет:

- подгружать ФИО, степень и звание научного руководителя или оппонента,
- контролировать выбор научного руководителя или оппонента только из одного списка,
- добавлять новую «персону», если в списках не был обнаружен искомый научный руководитель или оппонент.

К кнопке Сохранить добавлен обработчик события click, который осуществляет проверку о привязке сведений о защите к сотрудникам или подразделениям. Система выведет сообщение об ошибке, если не будет выполнено одно из условий:

1. хотя бы один из участников защиты должен быть выбран из списка сотрудников,
2. для хотя бы одного из участников защиты должно быть указано подразделение, в котором он работал на момент защиты.

В библиотеке реализованы вызовы callbacks для самостоятельной настройки дополнительных действий. Например, перед сохранением и обновлением информации о защите происходит обращение к функции `callback_before_insert_update` библиотеки `Xlaps_defense`. В данной функции происходит:

1. добавление новых записей в таблицу `default_xlaps_person`, если были добавлены новые «персоны»,
2. добавление записей в таблицы `default_xlaps_defense_members`, `default_xlaps_members_degree`, `default_xlaps_members_title`, если в защите были указаны научные руководители или оппоненты,
3. удаляет из массива `$post_array` поля, не содержащиеся в таблице `default_xlaps_defense`.

Ручной ввод информации весьма обременителен для пользователей, внушительный список полей ввода данных ведёт за собой большое количество

во опечаток. Для упрощения ввода данных о защите диссертаций, а так же для минимизации ошибок, весьма полезной оказалась интеграция возможности добавления сведений из файла автореферата диссертации (Рис. 14). Коллекция функций для анализа файла автореферата собрана в файле `pars_helper`.

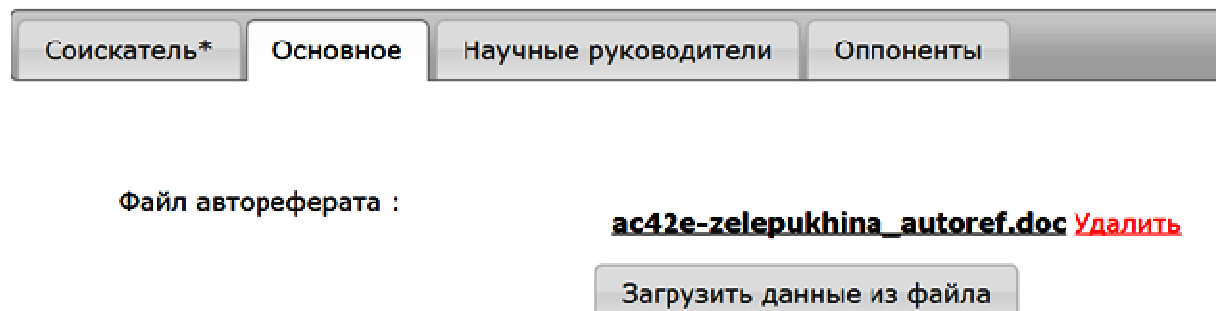


Рис. 14. Добавление файла автореферата

Нажатие на кнопку Загрузить данные из файла вызывает метод `ajax_parser` класса `Defenses`, в который передаётся имя загруженного файла методом `getJSON`. С помощью функции `parser` исходный текст автореферата преобразуется в структуру данных в виде ассоциативного массива (Array 1 на Рис. 15). Метод `for_pars` модели `xlaps_defense_m` преобразовывает некоторые поля полученного массива, например, вместо названий города и организации их идентификаторы (Array 2 на Рис. 15).

```
$city_title = iconv('windows-1251', 'UTF-8',  
$data['city']);  
  
$city = $this->db->from('default_xlaps_cities')  
->where('title', $city_title)  
->get()->row();
```

Метод `ajax_parser` преобразует полученный массив Array 2 в объект JSON [18] и возвращает клиенту.

```
function ajax_parser(){  
$this->load->helper('pars');  
$data = parser($_GET['file_name']);  
$result = $this->xlaps_defense_m->for_pars($data);
```

```
echo json_encode($result);
}
```

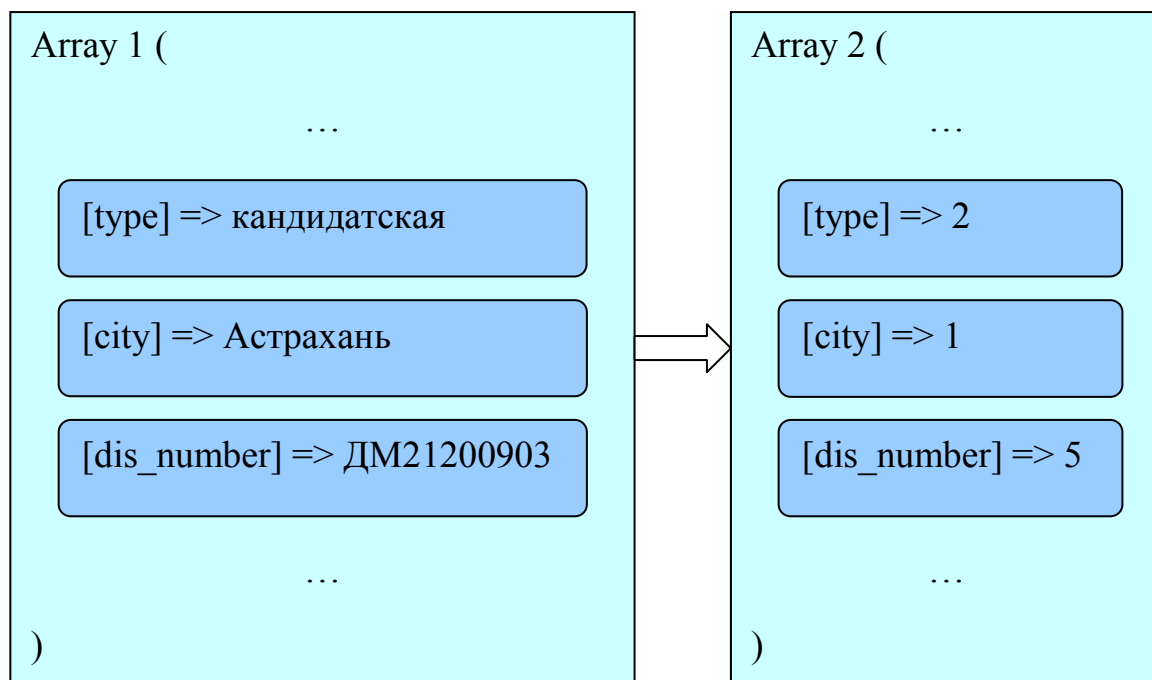


Рис. 15. Преобразование файла автореферата

Функция `fill_def_fields` заполняет поля формы полученными данными из файла автореферата:

```
function fill_def_fields (json, id, f) {
    SearchByLastName(json.student, "student");
    fill_field('defense_type_id', json.type);
    $('#field-defense_date').val(json.defense_date);
    $('#field-title').val(json.title);
    ...
    SearchByLastName(json.supervisors, "supervisors");
    SearchByLastName(json.opponents, "opponents");
}
```

В большинстве случаев во вкладке Основное автоматически будут заполнены практически все поля, за исключением полей Статус защиты и Дата утверждения, их придётся заполнить вручную.

Для заполнения вкладок соискатель, научные руководители и оппоненты происходит вызов функции `SearchByLastName` для идентификации связей участников защиты с сотрудниками и персонами. Для поиска похожих фамилий участников защиты среди сотрудников используется подпрограмма поиска нечётких дубликатов, реализованная в информационно-аналитической системе «Научная деятельность». А для поиска среди персон происходит вызов функции `search_person` класса `Defenses`, использующей полнотекстовый поиск:

```
function search_person($lname = null){
    $lname = urldecode ($lname);
    $data = $this->db->query('SELECT * FROM
`default_xlaps_person` WHERE MATCH (full_name) AGAINST
("'" . $lname . "')');'->result_array();
    echo json_encode($data);
}
```

Если в БД были найдены совпадения, то под соответствующими полями появятся ссылки на персональные страницы и кнопки со значком галочки, нажатие на которые подтверждает участие именно этого сотрудника в данной защите (Рис. 16).

Научный руководитель 1 Научный руководитель 2

Выбрать из списка сотрудников :

Выбрать из списка сотрудников

Тарасевич Юрий Юрьевич ✓

Рис. 16. Автоматическое заполнение вкладки Научный руководитель 1

Для обеспечения защищённого доступа пользователей к собственным сведениям о защитах и руководителей подразделений к сведениям своих сотрудников и дочерних подразделений в методе `_manage` при выполнении

операций редактирования, обновления и удаления данных производится проверка на принадлежность выбранного элемента коллекции текущему пользователю, при этом учитывая его роль в защите.

Если это сведения о собственных защитах, нужно проверить был ли в данной защите текущий пользователь соискателем, или же соискателем был кто-то из сотрудников пользователя:

```
case 'defense':  
if ( ($defense->student_id == ci()->current_user->id &&  
$defense->student_table == 'users') || ( $this->  
user_category == 'supervisor' && in_array($defense->  
student_id , $employee_ids ) ) || ci()->current_user->  
group == 'admin') {...} else{  
redirect('science/defenses/defense');  
}  
break;
```

Если это сведения о научном руководстве, необходимо проверить, был ли текущий пользователь или кто-то из сотрудников текущего пользователя научным руководителем в данной защите:

```
case 'guide':  
$sv = $this->xlaps_defense_m->  
get_supervisors($stateInfo->primary_key);  
$sv_ids = array();  
...  
if ( in_array(ci()->current_user->id , $sv_ids ) || ( $this->  
user_category == 'supervisor' &&  
sizeof(array_intersect($sv_ids , $employee_ids )) > 0 )  
|| ci()->current_user->group == 'admin') {...} else{  
redirect('science/defenses/guide');  
}  
break;
```

Функция `columns` библиотеки `grocery CRUD` позволяет определить поля, которые будут выводиться в списке записей:

```
$crud->columns('defense_type_id', 'defense_date',  
'specialties', 'organization');
```

Для работы со сведениями об обучении в аспирантуре/докторантуре создан класс `Education` по тому же принципу, что и `Defense` (Рис. 17).

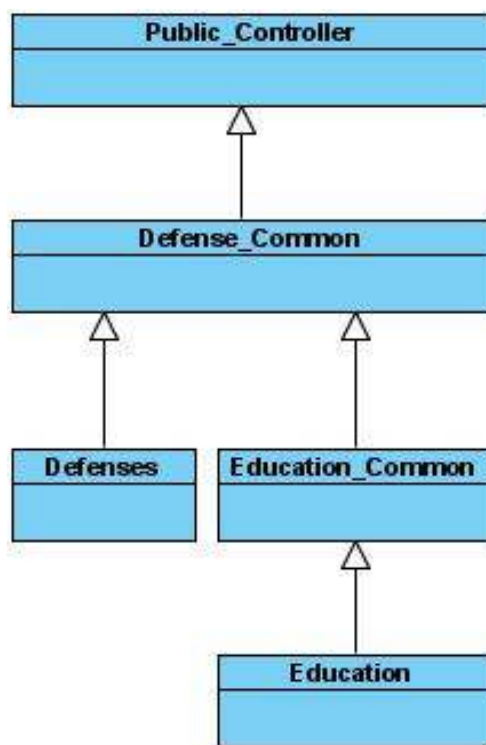


Рис. 17. Диаграмма классов

2.2. Программа для фильтрации/поиска информации об обучении/защитах

В подсистеме реализована возможность фильтрации и поиска записей по различным параметрам: год и тип защиты, сотрудники, подразделения. Данная возможность полезна при составлении отчетов, содержащих сведения о профессиональной квалификации, при выборе информации для отображения на персональных страницах сотрудников и подразделений, а также для установления связи между защитой и обучением.

В модели `xlaps_defense_m` реализован метод `get_items` (Рис. 18), позволяющий динамически генерировать запрос к БД согласно переданным в массиве `$search_params` параметрам:

- «`education_id`» — идентификатор обучения, для поиска защит возможно связанных с данным обучением,
- «`id`» — идентификатор защиты,
- «`type`» — тип диссертации (кандидатская, докторская),
- «`y`» — год защиты,
- «`role`» — роль участника защиты (соискатель, научный руководитель, оппонент),
- «`m`» — идентификаторы сотрудников (участников защиты),
- «`m_st`» — тип поиска по сотрудникам (хотя бы один, все выбранные, ни один из выбранных),
- «`t`» — принимает значения «`users`», «`person`»,
- «`d`» — идентификаторы подразделений,
- «`d_st`» — тип поиска по подразделениям (хотя бы один, ни один из выбранных),
- «`p`», «`limit`» — номер и число страниц, для пагинации.

Для поиска идентификаторов защит по сотрудникам и по подразделениям в модель `xlaps_defense_m` добавлены функции `get_by_members` и `get_by_departments`.

Метод `get_by_members` динамически генерирует запрос к БД, согласно переданным в массиве `$params_members` параметрам: «role», «m», «m_st», «t». Массив формируется в методе `get_items` при условии:

1. Если был передан параметр «m».

```
$params_members = array (  
    'role' => $search_params['role'],  
    'm' => $search_params['m'],  
    'm_st' => $search_params['m_st'],  
    't' => $search_params['t'],  
);
```

2. Если был передан параметр «d».

Интеграция информации о защите на уровень подразделения происходит через информацию о сотрудниках. В связи с этим для поиска защит по подразделениям необходимо найти всех сотрудников выбранных подразделений и дочерних и осуществить поиск защит по найденным сотрудникам. В системе реализованы методы для работы с отношениями Подразделение, Сотрудник, Подразделение – Сотрудник, посредством которых получаем массив `$members` с идентификаторами искомых сотрудников. Массив `$params_members` будет иметь вид:

```
$params_members = array (  
    'role' => $search_params['role'],  
    'm' => $members,  
    'm_st' => $search_params['d_st'],  
    't' => $search_params['t'],  
);
```

3. Если был передан параметр «education_id».

Идентификатор обучения передаётся в функцию для поиска защит возможно связанных с данным обучением. Метод `get_education` модели `xlaps_qualification_m` возвращает кортеж отношения `default_xlaps_education` (обучения), соответствующий переданному атрибуту в массив `$education` и массив `$params_members` формируется следующим образом:

```
$params_members = array (  
    'role' => 'student',  
    'm' => $education->student_id,  
    'm_st' => $ education ->student_table,  
    't' => $ education ->education_type_id)  
);
```

Наиболее важным является указание роли сотрудника: соискатель, научный руководитель, оппонент. В зависимости от значения этого параметра генерируются различные запросы к БД. Программа позволяет указать несколько сотрудников в предлагаемом списке и выбрать интересующий тип поиска:

1. **Хотя бы один** — найдёт те защиты, в которых указан хотя бы один из выбранных в списке сотрудников, согласно роли;
2. **Все выбранные** позволит найти защиты, в которых указаны все выбранные сотрудники, согласно роли;
3. **Ни один из** — найдёт те защиты, в которых не указан ни один из выбранных сотрудников.

Метод `get_by_departments` динамически генерирует запрос к БД, согласно переданным в массиве `$params_members` параметрам: «role», «d». Возвращает массив с идентификаторами защит, связь с которыми указана явно, через персон (информация хранится в отношении `default_xlaps_defense_member_department`). Массив `$params_members` формируется в методе `get_items`, при передаче параметра «d».

Идентификаторы защит, полученные из метода `get_by_members` в массив `$ids_by_members`, автоматически добавляются к массиву `$search_params['id']`:

```
$search_params['id'] =  
array_merge($search_params['id'], $ids_by_members);
```

В зависимости от значения, переданного в функцию `get_items`, параметра «`d_st`»: хотя бы один, ни один из выбранных, идентификаторы, полученные из метода `get_by_departments`, добавляются или исключаются из массива `$search_params['id']`.

После этого, если массив `$search_params['id']` не пуст, генерируется оператор `WHERE` для поиска защит по идентификатору:

```
$this->db->where_in('default_xlaps_defense.id',  
$search_params['id']);
```

Осуществление поиска защит по таким параметрам как тип или год защиты не представляет трудности, нужно лишь проверить, был ли передан соответствующий параметр, и если да с помощью функции `$this->db->where()` задать оператор `WHERE`:

```
if (sizeof($search_params['y']) > 0)  
$this->db->where_in  
('YEAR(default_xlaps_defense.defense_date)',  
$search_params['y']);
```

Некоторые запросы к БД выразим с помощью выражений реляционной алгебры [19].

Выпишем реляционную модель данных. Введём обозначения атрибутов всех необходимых для составления запросов сущностей. Пусть *A1* — код защиты, *A2* — код соискателя, *A3* — таблица соискателя, *A4* — код участника защиты (в данном случае участник защиты — научный руководитель или оппонент), *A5* — таблица участника защиты, *A6* — роль участника защиты, *A7* — код подразделения. Обозначим схему отношения ЗАЩИТА как *R1*,

ЗАЩИТА – УЧАСТНИК как $R2$, ЗАЩИТА – УЧАСТНИК – ПОДРАЗДЕЛЕНИЕ как $R3$. Тогда реляционная модель описывается следующей совокупностью схем отношений:

$R1 (A1, A2, A3)$

$R2 (A1, A4, A5, A6)$

$R3 (A1, A4, A5, A6, A7)$

Отношения, соответствующие вышеуказанным схемам, будем обозначать: $r1, r2, r3$.

Определим некоторые операции:

1. Проекция. $\pi_{A1, A2, \dots, Am}(r)$.

Проекция $\pi_{A1, A2, \dots, Am}(r)$ — множество кортежей, получаемых из кортежей отношения r выбором столбцов с именами $A1, A2, \dots, Am$.

2. Выбор (селекция) $\sigma_F(r)$.

Пусть F — формула, образованная операндами, являющимися константами или именами атрибутов, арифметическими операторами сравнения, логическими операторами (и, или, не). Тогда выбором (селекцией) σ_F называется множество кортежей, компоненты которого удовлетворяют условию, заданному формулой F .

3. Пересечение $r \cap s$

Пересечением отношений r и s называется множество кортежей, принадлежащих как r , так и s .

Приведем ряд примеров составления запросов для фильтра с помощью формальных операций для реляционной модели, рассмотренной выше.

Пример 1.

Выбрать список кодов тех защит, соискателем в которых являлся хотя бы один из выбранных сотрудников. Список переданных параметров имеет

вид: m — идентификаторы выбранных сотрудников, t принимает значение «users», $role$ — «student».

Рассмотрим отношение $r1$. Формула для искомых кортежей будет иметь вид:

$$F:(A2)=m \text{ И } (A3)=t.$$

Выберем защиты, удовлетворяющие условиям, заданным формулой F :

$$\sigma_F(r1).$$

Затем возьмём проекцию полученного отношения на атрибут $A1$:

$$\pi_{A1}(\sigma_F(r1)).$$

Получаем требуемый результат.

Пример 2.

Выбрать список кодов тех защит, научным руководителем в которых не являлся ни один из выбранных сотрудников. Список переданных параметров имеет вид: m — идентификаторы выбранных сотрудников, t принимает значение «users», $role$ — «supervisor».

Сначала из отношения $r2$ выберем кортежи, удовлетворяющие условию, заданному формулой:

$$F1:HE((A4)=m \text{ И } (A5)=t \text{ И } (A6)=role).$$

Обозначим полученное отношение $rp1$. (Дальнейшие промежуточные отношения будем обозначать последовательно $rp1$, $rp2$, $rp3$ и т.д.).

$$rp1 = \sigma_{F1}(r2).$$

Далее нас будет интересовать только атрибут $A1$ — "код защиты". Поэтому возьмем проекцию на этот столбец:

$$rp1 = \pi_{A1}(\sigma_{F1}(r2)).$$

Возьмём проекцию отношения $r1$ на столбец $A1$:

$$rp2 = \pi_{A1}(r1).$$

Используя операцию «пересечение» для отношений $rp1$ и $rp2$ получим требуемый результат:

$$rp3 = rp1 \cap rp2.$$

Пример 3.

Выбрать список кодов тех защит, участники которых относились к выбранным подразделениям. Список переданных параметров имеет вид: d — идентификаторы выбранных подразделений и всех дочерних, $role$ принимает значения «student», «supervisor» или «opponent».

Сначала из отношения $r3$ выберем кортежи, удовлетворяющие условию, заданному формулой:

$$F2: (A6) = role \text{ И } (A7) = d.$$

Обозначим полученное отношение $rp4$, и возьмем проекцию на столбец $A1$ — "код защиты".

$$rp4 = \pi_{A1}(\sigma_{F2}(r3)).$$

Используя операцию «пересечение» для отношений $rp4$ и $rp2$ из предыдущего примера получим требуемый результат:

$$rp5 = rp4 \cap rp2.$$

Указанные пользователем системы параметры на форме поиска защит диссертаций методом POST передаются в метод `search` класса `Defenses`. Метод `search` так же используются для вывода сведений о профессиональной квалификации на персональные странички сотрудников и подразделений, в этом случае данные передаются по AJAX [20]. Функция `$this->input->is_ajax_request()` позволяет проверить, каким образом были переданы параметры: методом POST или по AJAX, в последнем случае выводим список защит на странице подразделения или сотрудника без формы фильтрации.

Метод `get_by_params` модели возвращает полную информацию обо всех найденных защитах. Для каждой защиты представлена основная информация, которая получена методом `defense_main_to_html` из библиотеки `xlaps_science`, и дополнительная — метод `defense_extra_to_html`.

2.3. Рейтинг диссертаций

Основное содержание диссертации отражено в публикациях, в связи с этим рейтинг диссертации будем получать суммированием рейтингов всех статей, относящихся к данной работе.

Для оценки статьи в журнале за основу возьмём следующую формулу:

$$R = IF \times \frac{NP}{TNP \times NCA},$$

где

IF — импакт-фактор журнала (Impact Factor) [21],

NP — число страниц в статье (Number of Pages),

TNP — типичное число страниц в статье (Typical Number of Pages),

NCA — число соавторов (Number of Co-Authors).

Для журналов, не имеющих импакт-фактора, но входящих в «Перечень российских рецензируемых научных журналов, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук» [22], примем импакт-фактор равным 0,01, для прочих журналов — 0,001.

Число страниц в статье (*NP*) и число соавторов (*NCA*) легко получить из соответствующих атрибутов отношений БД (отношения `default_xlaps_publications`, `default_xlaps_publications_authors`). Типичное число страниц в статье (*TNP*) примем равным 9, ориентируясь на журнал *Physical Review* [23].

Расчет осложняется тем, что данные об импакт-факторах журналов не могут быть получены автоматически из базы данных Web of Science (WOS)

[24] из-за отсутствия у Астраханского государственного университета подписки к продуктам компании Thomson Reuters [25].

Апробация методики расчета рейтинга проводилась на основе данных 15 авторефератов кандидатских диссертаций по физико-математическим, техническим и педагогическим наукам. Результаты расчета отражены в *Таблица 5*.

Возможны вариации метода расчета рейтинга диссертаций:

1. Учитывать только те публикации, в которых соискатель является одним из трёх первых соавторов. Результаты приведены в *Таблица 6* приложения;
2. Учитывать только те публикации, в которых соискатель является первым соавтором (*Таблица 7* приложения);
3. Учитывать публикации, в которых соискатель является одним из трёх первых соавторов, причём рейтинг делить в пропорции 50%-30%-20% для первого, второго, третьего соавтора соответственно (*Таблица 8* приложения).

Таблица 5

Рейтинг диссертаций

№ диссертации	рейтинг	отрасль науки	дис. совет	статьи в журналах с импакт-фактором		статьи ВАК		прочие	
				число	рейтинг	кол-во	рейтинг	кол-во	рейтинг
1	4,722	физико-математические	ЮФУ	3	4,711	2	0,011	0	0
2	3,846	физико-математические	ЮФУ	2	3,84	1	0,006	0	0
3	1,361	физико-математические	АГУ	2	1,347	2	0,014	0	0
4	1,012	физико-математические	АГУ	3	1,009	1	0,003	0	0
5	0,557	физико-математические	АГУ	2	0,553	1	0,004	0	0
6	0,316	физико-математические	АГУ	2	0,312	1	0,004	1	0
7	0,07	физико-математические	АГУ	1	0,059	3	0,011	0	0
8	0,034	физико-математические	АГУ	0	0	12	0,034	0	0
9	0,028	физико-математические	АГУ	0	0	12	0,028	1	0
10	0,023	физико-математические	АГУ	0	0	9	0,023	2	0
11	0,008	технические	АГУ	0	0	3	0,008	1	0
12	0,005	технические	АГУ	0	0	2	0,005	1	0
13	0,003	физико-математические	ЯрГУ	0	0	1	0,002	2	0,001
14	0,002	педагогические	АГУ	0	0	1	0,002	0	0
15	0	педагогические	МПУ	0	0	0	0	0	0

ЗАКЛЮЧЕНИЕ

Реализован программный модуль «Подготовка кадров высшей квалификации», который является относительно независимой частью информационно-аналитической системы «Программа для сбора и хранения научной и наукометрической информации». Программный модуль ориентирован на использование в организациях, осуществляющих подготовку кадров высшей квалификации (университеты, исследовательские центры и т.п.) и предназначен для сбора и хранения информации о профессиональной квалификации научных и научно-педагогических кадров.

Основные функциональные возможности программы:

Все пользователи, включая незарегистрированных в системе, имеют возможность 1) просмотра сведений о профессиональной квалификации пользователей системы; 2) поиска и фильтрации информации о защитах; 3) поиска и фильтрации информации об аспирантуре, докторантуре.

Зарегистрированный пользователь (сотрудник организации), кроме того, имеет дополнительные возможности: 1) добавление, редактирование и удаление сведений о своей профессиональной квалификации; 2) создание связей между сведениями о профессиональной квалификации с пользователями системы, согласно их роли: соискатель, научный руководитель, оппонент; 3) создание связи между информацией об обучении в аспирантуре, докторантуре и о защите диссертации; 4) создание связей между сведениями об обучении/защите с подразделениями организации.

Руководитель подразделения, кроме того, имеет возможность добавления, редактирования и удаления сведений о профессиональной квалификации сотрудников своих подразделений.

Для облегчения работы с подсистемой подготовлено «Руководство пользователя» [26]. Опубликованы научные работы [27, 28, 29].

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Интеллектуальная Система Тематического Исследования НАучно-технической информации [сайт]. URL: <http://istina.imec.msu.ru/> (дата обращения: 08.10.2012)
2. Информационно-аналитическая система сопровождения научно-исследовательской деятельности СПбГУ [сайт]. URL: <https://ias.csr.spbu.ru/> (дата обращения: 08.10.2012)
3. Публикации ученых БелГУ [Электронный ресурс] // Белгородский государственный университет [сайт]. URL: <http://unid.bsu.edu.ru/unid/res/publ/> (дата обращения: 08.10.2012)
4. Edinburgh Research Explorer [сайт]. URL: <http://www.research.ed.ac.uk/portal/en/> (дата обращения: 08.10.2012)
5. Астраханский государственный университет. Научная деятельность [сайт]. URL: <http://science.aspu.ru> (дата обращения: 08-10-2012)
6. Зелепухина В.А., Тарасевич Ю.Ю. Программа для сбора и хранения научной и наукометрической информации// Программа для ЭВМ, свидетельство № 2012612861 от 22.03.2012 г.
7. Зелепухина В. А., Тарасевич Ю.Ю. Концепция информационно-аналитической системы для сбора и анализа научной и наукометрической информации в организации // Информатизация образования и науки.— 2013.— Вып. 2(18).— С. 133-144
8. Положение о квалификационных категориях, квалификационных и профессиональных требованиях к профессорско-преподавательскому составу
9. EuroCRIS | Research Information | CERIF | LOD | CRIS [сайт]. URL: <http://www.eurocris.org/Index.php?page=featuresCERIF&t=1> (дата обращения: 08.10.2012)

10. Кузнецов С. Д. Основы баз данных: учебное пособие // С. Д. Кузнецов — 2-е изд., испр. — М.: Интернет-Университет Информационных Технологий; БИНОМ Лаборатория знаний, 2007. — 484 с., ил.

11. Фаулер М. UML. Основы, 3-е издание. — Пер. с английского. — СПб: Символ-Плюс, 2004. — 192 с., ил.

12. Приказ Министерства образования и науки РФ от 25 февраля 2009 г. N 59 "Об утверждении Номенклатуры специальностей научных работников" (с изменениями и дополнениями) // Бюллетень нормативных актов федеральных органов исполнительной власти от 18 мая 2009 г. N 20

13. Документация по PHP5 [Электронный ресурс] — URL: <http://php.net/>, свободный

14. Документация по СУБД MySQL 5.1 [Электронный ресурс] — URL: <http://www.mysql.com/>, свободный

15. Документация по CodeIgniter 2.0 [Электронный ресурс] — URL: <http://ellislab.com/codeigniter>, свободный

16. Документация по grocery CRUD 1.3.3 [Электронный ресурс] — URL: <http://www.grocerycrud.com/>, свободный

17. Документация по jQuery [Электронный ресурс] — URL: <http://jquery.com/>, <http://jqueryui.com/>, свободный

18. Документация по JSON [Электронный ресурс] — URL: <http://json.org/>, свободный

19. Грей П. Логика, алгебра и базы данных. — М.: Машиностроение, 1989. — С. 188-213. — 368 с.

20. Освоение AJAX: Часть 1. Введение в AJAX [Электронный ресурс] — URL: <http://www.ibm.com/developerworks/ru/library/wa-ajaxintro1/index.html>, свободный

21. Thomson Reuters | The Thomson Reuters Impact Factor | Science [сайт]. URL:

http://thomsonreuters.com/products_services/science/free/essays/impact_factor/
(дата обращения 17.05.2013)

22. Перечень российских рецензируемых научных журналов [сайт].
URL: http://vak.ed.gov.ru/ru/help_desk/list/ (дата обращения 17.05.2013)

23. Physical Review E [сайт]. URL: <http://pre.aps.org/> (дата обращения 17.05.2013)

24. Thomson Reuters | Web of Science | Science [сайт].
URL: http://thomsonreuters.com/products_services/science/science_products/a-z/web_of_science/ (дата обращения: 17.05.2013)

25. Thomson Reuters [сайт]. URL: <http://thomsonreuters.com/> (дата обращения 17.05.2013)

26. Данилова Т.С. Информационно-аналитическая система для сбора, хранения и анализа научной и наукометрической информации. Руководство пользователя. Часть 2. Подсистема «Подготовка кадров высшей квалификации». — Астрахань: ООО «Типография Новая Линия», 2013. — ISBN 978-5-901918-67-8

27. Бурмистров А.С., Данилова Т.С., Сальшин В.И., Умаров А.С., Зелепухина В.А., Тарасевич Ю.Ю. Особенности разработки информационной системы для сбора и хранения результатов научной деятельности // ДВАДЦАТАЯ МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ. МАТЕМАТИКА. КОМПЬЮТЕР. ОБРАЗОВАНИЕ. г. Пущино, 28 января 2 февраля 2013 г. Тезисы — М., Ижевск: РХД, 2013. — С. 205

28. Данилова Т.С., Зелепухина В.А., Тарасевич Ю.Ю. Концепция разработки базы данных «Подготовка кадров и профессиональная квалификация» // ДВАДЦАТАЯ ВСЕРОССИЙСКАЯ НАУЧНО-МЕТОДИЧЕСКАЯ КОНФЕРЕНЦИЯ ТЕЛЕМАТИКА 2013. Санкт-Петербург, 2013 г.

29. Тарасевич Ю.Ю., Зелепухина В.А., Данилова Т.С., Бурмистров А.С., Сальшин В.И., Умаров А.С., Бичарова М.М. Информационно-

аналитическая система «Результаты научной деятельности» // Программа конференции SCIENCE ONLINE XVII [Электронный ресурс] — URL: http://elibrary.ru/projects/conference/italy2013/program_so_2013.pdf (дата обращения 22.05.2013)

ПРИЛОЖЕНИЯ

Таблица 6

Рейтинг диссертаций с учетом первых трёх соавторов

№ диссертации	рейтинг	отрасль науки	дис. совет	статьи в журналах с импакт-фактором		статьи ВАК		прочие	
				число	рейтинг	кол-во	рейтинг	кол-во	рейтинг
1	4,722	физико-математические	ЮФУ	3	4,711	2	0,011	0	0
2	3,846	физико-математические	ЮФУ	2	3,84	1	0,006	0	0
3	1,263	физико-математические	АГУ	2	1,249	2	0,014	0	0
4	1,012	физико-математические	АГУ	3	1,009	1	0,003	0	0
5	0,557	физико-математические	АГУ	2	0,553	1	0,004	0	0
6	0,316	физико-математические	АГУ	2	0,312	1	0,004	1	0
7	0,067	физико-математические	АГУ	1	0,059	3	0,008	0	0
8	0,032	физико-математические	АГУ	0	0	12	0,032	0	0
9	0,023	физико-математические	АГУ	0	0	12	0,023	1	0
10	0,023	физико-математические	АГУ	0	0	9	0,023	2	0
11	0,008	технические	АГУ	0	0	3	0,008	1	0
12	0,005	технические	АГУ	0	0	2	0,005	1	0
13	0,003	физико-математические	ЯрГУ	0	0	1	0,002	2	0,001
14	0,002	педагогические	АГУ	0	0	1	0,002	0	0
15	0	педагогические	МПУ	0	0	0	0	0	0

Таблица 7

Рейтинг диссертаций с учетом первого соавтора

№ диссертации	рейтинг	отрасль науки	дис. совет	статьи в журналах с импакт-фактором		статьи ВАК		прочие	
				число	рейтинг	кол-во	рейтинг	кол-во	рейтинг
1	1,428	физико-математические	ЮФУ	1	1,424	1	0,004	0	0
3	1,263	физико-математические	АГУ	2	1,249	2	0,014	0	0
6	0,316	физико-математические	АГУ	2	0,312	1	0,004	1	0
5	0,217	физико-математические	АГУ	2	0,213	1	0,004	0	0
8	0,011	физико-математические	АГУ	0	0	12	0.011	0	0
7	0,006	физико-математические	АГУ	1	0	3	0,006	0	0
10	0,006	физико-математические	АГУ	0	0	9	0,006	2	0
12	0,005	технические	АГУ	0	0	2	0,005	1	0
11	0,003	технические	АГУ	0	0	3	0,003	1	0
13	0,003	физико-математические	ЯрГУ	0	0	1	0,002	2	0,001
14	0,002	педагогические	АГУ	0	0	1	0,002	0	0
2	0	физико-математические	ЮФУ	0	0	0	0	0	0
4	0	физико-математические	АГУ	3	0	1	0	0	0
9	0	физико-математические	АГУ	0	0	12	0	1	0
15	0	педагогические	МПУ	0	0	0	0	0	0

Таблица 8

Рейтинг диссертаций с учетом первых трех соавторов в пропорции 50%-30%-20%

№ диссертации	рейтинг	отрасль науки	дис. совет	статьи в журналах с импакт-фактором		статьи ВАК		прочие	
				число	рейтинг	кол-во	рейтинг	кол-во	рейтинг
1	4,905	физико-математические	ЮФУ	3	4,891	2	0,014	0	0
2	3,347	физико-математические	ЮФУ	2	3,342	1	0,005	0	0
3	1,263	физико-математические	АГУ	2	1,249	2	0,014	0	0
4	0,708	физико-математические	АГУ	3	0,706	1	0,002	0	0
5	0,702	физико-математические	АГУ	2	0,697	1	0,005	0	0
6	0,242	физико-математические	АГУ	2	0,239	1	0,003	1	0
7	0,062	физико-математические	АГУ	1	0,053	3	0,009	0	0
8	0,027	физико-математические	АГУ	0	0	12	0,027	0	0
10	0,022	физико-математические	АГУ	0	0	9	0,022	2	0
9	0,014	физико-математические	АГУ	0	0	12	0,014	1	0
12	0,01	технические	АГУ	0	0	2	0,01	1	0
11	0,007	технические	АГУ	0	0	3	0,007	1	0
14	0,004	педагогические	АГУ	0	0	1	0,004	0	0
13	0,003	физико-математические	ЯрГУ	0	0	1	0,002	2	0,001
15	0	педагогические	МПУ	0	0	0	0	0	0