

УДК 004.6

А.А. Козинский, А.А. Гуносов

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ ДЛЯ ПРОГРАММНОЙ СИСТЕМЫ РАСЧЕТА ОБЩЕЙ НАГРУЗКИ КАФЕДРЫ

Статья посвящена описанию модели данных для программной системы расчета общей нагрузки кафедры высшего учебного заведения. Приводятся основные этапы и порядок формирования общей нагрузки кафедры. Рассматривается структура и особенности реализации основных элементов разработанной модели данных. Описываются некоторые особенности процесса преобразования логической модели данных в физическую средствами MySQL Workbench. Настоящая статья также содержит фрагмент программного кода на языке SQL для создания схемы базы данных. Актуальность разработки модели объясняется значительным объемом работ для расчета общей нагрузки кафедры и других задач автоматизации управления подразделениями вуза. Эффективность такой работы зависит от степени использования современных информационных технологий.

Введение

Одним из направлений деятельности высшего учебного заведения является управление образовательными процессами. Один из этапов управления – планирование учебной нагрузки. Это многоэтапная процедура, которая проводится перед началом каждого учебного года. В процедуру планирования вовлечено множество подразделений вуза, таких, как ректорат, учебно-методический отдел, деканаты и кафедры. Большая часть планирования осуществляется на кафедрах, выполняющих учебную нагрузку. Расчет общей нагрузки кафедры обычно производится с помощью электронных таблиц Microsoft Excel или другими способами. Особенность таких расчетов в том, что они носят изолированный характер. Данные необходимые для расчетов хранятся в той же таблице с расчетами и не могут быть использованы в других подразделениях. В силу того, что данные, как правило, вносятся в файлы электронных таблиц вручную, процесс расчета является трудоемким и занимает продолжительное время. Выходом из этой ситуации может стать создание программной системы для автоматизации процесса расчета общей нагрузки кафедры, которая в будущем должна использоваться на различных этапах управления (планирование учебного процесса, анализ результатов и т.п.).

В настоящей статье описана реляционная модель данных для программной системы расчета общей нагрузки кафедры, которая в дальнейшем будет использоваться на различных этапах управления учебным процессом.

Описание модели данных

Созданию модели данных предшествовал анализ предметной области. Объектом исследования являлась процедура планирования образовательных процессов. Предмет исследования – расчет общей нагрузки кафедры.

Согласно документированной процедуры ДП 7.3.0-2012 системы менеджмента качества [1, с. 17], для расчета объема учебной работы на кафедрах учебно-методический отдел (УМО) разрабатывает «Нормы времени университета для расчета объема учебной нагрузки профессорско-преподавательского состава на учебный год» в соответствии с «Примерными нормами времени для расчета объема учебной работы и основных видов учебно-методической, научно-исследовательской и других работ, выполняемых профессорско-преподавательским составом высших учебных заведений».

Расчет учебной нагрузки осуществляется на основании [1, с. 17]:

– рабочих учебных планов;

- сведений о составе студентов, количестве групп и подгрупп на учебный год;
- норм времени университета для расчета объема учебной нагрузки профессорско-преподавательского состава на учебный год;
- «Положения об организации самостоятельной работы студентов»;
- решения ректората и постановления Совета Университета.

На основе проведенного анализа было выполнено построение реляционной модели данных. Реляционная модель данных – это теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теории множеств и логика первого порядка [2]. Таким образом, созданная модель является математической моделью данной предметной области.

Были созданы сущности `qualifications` (квалификации), `forms_of_education` (формы обучения), `types_of_standards` (типы стандартов), `types_of_plans` (типы планов). Атрибуты и экземпляры созданных сущностей представлены на рисунке 1.

| qualifications | | forms_of_education | |
|----------------|-----------------------|--------------------|----------------|
| id | description | id | Description |
| 1 | Математик-программист | 1 | Очная дневная |
| 2 | Математик-экономист | 2 | Заочная |
| 3 | Физик. Программист | 3 | Очная вечерняя |

| types_of_standards | | types_of_plans | |
|--------------------|---------------------------|----------------|-------------|
| id | description | id | Description |
| 1 | Специальность | 1 | Базовый |
| 2 | Направление специальности | 2 | Рабочий |
| 3 | Специализация | | |

Рисунок 1 – Сущности `qualifications`, `forms_of_education`, `types_of_standards`, `types_of_plans`

Сущность `specializations` (специализации) предназначена для хранения информации о специальностях, направлениях специальностей, специализации. В качестве ключа используется код специальности/направления специальности/специализации, т.к. данный атрибут полностью соответствует требованиям, предъявляемым к первичному ключу. Сущность `level_of_education` (уровень образования) хранит информацию об уровнях образования. Структура сущностей `specializations` и `level_of_education` представлена на рисунке 2.

В сущности `directions_of_specialization` (направления специализации) вместо текстовых значений квалификации, типа стандарта, специальности и уровня образования используются внешние ключи на соответствующие характеристические сущности, что исключает излишнее дублирование данных. Одним из внешних ключей сущности `directions_of_specialization` является атрибут `directions_of_specialization_id`. Он является указателем на идентификатор (`id`) собственной сущности и служит для организации цепочек подчиненности: специальность → направление специальности → специализация. Структура сущности показана на рисунке 3.

| specializations | | forms_of_education | |
|---------------------|--|--------------------|--|
| code_specialization | description | id | description |
| 1-31 03 06 | Экономическая кибернетика (по направлениям) | 1 | Незаконченное высшее (не менее двух лет обучения в вузе) |
| 1-31 03 06-01 | Экономическая кибернетика (математические методы и компьютерное моделирование в экономике) | 2 | Образовательная программа бакалавра |
| 1-31 03 06-01 06 | Математические методы и информационные технологии в логистике | 3 | Образовательная программа специалиста |
| 1-31 03 06-01 04 | Информационные технологии управления в экономике | | |

Рисунок 2 – Сущности specializations, level_of_education

Например, из данных, приведенных на рисунке 3, следует, что существует специальность «Экономическая кибернетика», направление специальности «Математические методы и компьютерное моделирование в экономике» и данное направление имеет две специализации: «Математические методы и информационные технологии в логистике» и «Информационные технологии управления в экономике».

| id | qualifications id | types of standards id | specializations code | level of education id | directions of specialization id |
|----|-------------------|-----------------------|----------------------|-----------------------|---------------------------------|
| 1 | 2 | 1 | 1-31 03 06 | 3 | NULL |
| 2 | 2 | 2 | 1-31 03 06-01 | 3 | 1 |
| 3 | 2 | 3 | 1-31 03 06-01 06 | 3 | 2 |
| 4 | 2 | 3 | 1-31 03 06-01 04 | 3 | 2 |

Рисунок 3 – Сущность directions_of_specialization

В рабочих учебных планах также присутствует информация о факультетах и кафедрах. Кафедры могут как входить в состав какого-либо факультета, так и быть общеуниверситетскими. Выполнять нагрузку могут не только кафедры, но и некоторые подразделения вуза. Для хранения информации о подразделениях вуза, их иерархии и составе выполняемых функций созданы сущности: divisions (отделы), property_divisions (свойства отделов), characteristics_divisions (характеристики отделов). Для сущности divisions используется такой же подход как и для сущности directions_of_specialization с целью организации цепочек подчиненности. На основе атрибута divisions_id выстраивается иерархия подразделений, например: Брестский государственный университет имени А.С. Пушкина → Физико-математический факультет → Кафедра математического моделирования. Сущности property_divisions и characteristics_divisions служат для определения свойств отделов. Пример структуры и хранимых данных представлен на рисунке 4.

| divisions | | | |
|-----------|-------------------------|---|--------------|
| id | short description | description | divisions id |
| 1 | БрГУ имени А.С. Пушкина | Брестский государственный университет имени А.С. Пушкина | |
| 2 | ФМФ | Физико-математический факультет | 1 |
| 3 | ПМТП | Кафедра прикладной математики и технологий программирования | 2 |
| 4 | ММ | Кафедра математического моделирования | 2 |

| characteristics_divisions | |
|---------------------------|-----------------------|
| divisions id | property divisions id |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 3 |

| property_divisions | | |
|--------------------|-------------|----------------------------|
| id | description | explanation |
| 1 | Университет | Университет |
| 2 | Факультет | Отдел является факультетом |
| 3 | Кафедра | Отдел является кафедрой |

Рисунок 4 – Сущности divisions, characteristics_divisions, property_divisions

Для хранения данных об учебных планах и дисциплинах учебного плана созданы следующие сущности:

- types_of_plans (типы планов) – содержит информацию о типе учебных планов, например, базовый, рабочий;
- plans (планы) – используется для хранения основной информации об учебном плане (факультет, выпускающая кафедра, учебный год, специальность, курс и др.);
- components (компоненты) – используется для хранения информации о видах компонентов (обязательный, вузовский и т.п.);
- cycles_of_disciplines (цикл дисциплин) – содержит информацию о циклах дисциплин (основные, социально-гуманитарные, дисциплины специализации);
- disciplines (дисциплины) – содержит информацию о полных и кратких наименованиях дисциплин (программирование, методы численного анализа);
- lines_plans (строки планов) – включает информацию о дисциплинах учебного плана, такую, как ссылка на учебный план, ссылка на цикл дисциплин, ссылка на наименование дисциплины и др.;
- content_lines_plans (содержание строк планов) – хранит информации о распределении учебной нагрузки дисциплины по семестрам.

Также создан ряд сущностей для хранения данных о студенческих группах, подгруппах и лекционных потоках.

Реализация модели данных

Проектирование реляционной модели данных проводилось в приложении MySQL Workbench. Кроме средств для создания логической модели данных, данное приложение обладает возможностью визуального проектирования физической модели [3]. Физическая модель данных проектировалась исходя из предполагаемого объема

хранимой информации. Данный объем в первую очередь зависит от количественного состава студентов и преподавателей. В Брестском государственном университете имени А.С. Пушкина количество студентов составляет около 9 тысяч и около 600 преподавателей [4].

На этапе физического моделирования для каждого атрибута выбран соответствующий тип данных. Типы данных выбирались исходя из допустимых диапазонов значений. «Щедрость» при назначении типа данных увеличивает размер записи, что впоследствии может привести к замедлению поиска необходимой записи в базе данных, т.к. записи хранятся на диске последовательно. Для большинства таблиц выбран тип хранилища InnoDB. Основным отличием InnoDB от других типов хранилищ СУБД MySQL является наличие механизма транзакций и внешних ключей [5]. Для текстовых полей установлена кодировка UTF-8. Также на этапе физического моделирования к некоторым полям добавлялись индексы. Как правило, поля которые должны быть проиндексированы выбираются на этапе оптимизации запросов к базе данных [6, с. 273]. Однако MySQL Workbench автоматически добавляет индексы к первичным ключам, ключам-кандидатам и внешним ключам. На рисунке 5 представлен фрагмент схемы физической модели данных.

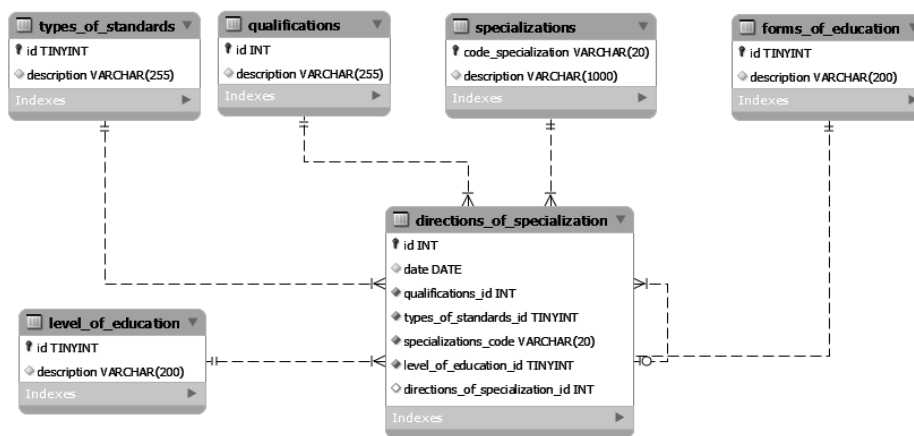


Рисунок 5 – Фрагмент схемы физической модели данных

Фрагмент программного кода на языке SQL для создания схемы базы данных приведен ниже:

Фрагмент программного кода

```
CREATE TABLE IF NOT EXISTS `webinedub`.`divisions` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `short_description` VARCHAR(45) NOT NULL COMMENT 'краткое наименование',
  `description` VARCHAR(255) NOT NULL COMMENT 'наименование',
  `divisions_id` INT UNSIGNED NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_divisions_divisions_idx` (`divisions_id` ASC),
  CONSTRAINT `fk_divisions_divisions`
  FOREIGN KEY (`divisions_id`) REFERENCES `webinedub`.`divisions` (`id`)
  ON DELETE RESTRICT ON UPDATE RESTRICT) ENGINE = InnoDB COMMENT = 'отделы';
CREATE TABLE IF NOT EXISTS `webinedub`.`property_divisions` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `description` VARCHAR(45) NOT NULL COMMENT 'наименование',
  `explanation` VARCHAR(255) NOT NULL COMMENT 'примечание\n',
  PRIMARY KEY (`id`),
  UNIQUE INDEX `descriptions_UNIQUE` (`description` ASC)) ENGINE = InnoDB COMMENT = 'свойства отделов';
```

```
CREATE TABLE IF NOT EXISTS `webinedudb`.`characteristics_divisions` (  
  `divisions_id` INT UNSIGNED NOT NULL AUTO_INCREMENT COMMENT 'id отдела',  
  `property_divisions_id` INT UNSIGNED NOT NULL COMMENT 'id свойства отдела',  
  PRIMARY KEY (`divisions_id`, `property_divisions_id`),  
  INDEX `fk_characteristics_divisions_property_divisions_idx` (`property_divisions_id` ASC),  
  CONSTRAINT `fk_characteristics_divisions_divisions`  
    FOREIGN KEY (`divisions_id`) REFERENCES `webinedudb`.`divisions` (`id`)  
    ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `fk_characteristics_divisions_property_divisions`  
    FOREIGN KEY (`property_divisions_id`) REFERENCES `webinedudb`.`property_divisions` (`id`)  
    ON DELETE RESTRICT ON UPDATE RESTRICT) ENGINE = InnoDB COMMENT = 'характеристика  
отделов';
```

Заклучение

В настоящей статье представлено описание реляционной модели данных для программной системы расчета общей нагрузки кафедры. Приведен фрагмент программного кода для создания схемы базы данных. Модель данных универсальна и разработана для решения задач управления подразделением вуза. В числе актуальных задач, для решения которых будет применена модель: управление нагрузкой преподавателей, автоматизация управлением расписанием занятий, отчеты о выполнении запланированной нагрузки и др. Все перечисленные задачи должны опираться на созданную базу данных. Широкий круг задач налагает повышенные требования ко всем уровням моделирования. Такие требования выполнены для представленных в настоящей работе результатов.

СПИСОК ЛИТЕРАТУРЫ

1. Проектирование образовательных программ и планирование образовательных процессов [Тест]: ДП 7.3.0-2012 1. – Введ. 2002-05-07.
2. Реляционная база данных – Википедия [Электронный ресурс] Режим доступа: http://ru.wikipedia.org/wiki/Реляционная_база_данных. – Дата доступа: 04.03.2014.
3. MySQL : MySQL Workbench 6.0 [Электронный ресурс] Режим доступа: <http://www.mysql.com/products/workbench>. – Дата доступа: 04.03.2014.
4. Общие сведения | Брестский государственный университет имени А.С. Пушкина [Электронный ресурс] Режим доступа: <http://www.brsu.by/univer/obshchie-svedeniya>. – Дата доступа: 04.03.2014.
5. InnoDB – Википедия [Электронный ресурс] Режим доступа: <http://ru.wikipedia.org/wiki/InnoDB>. – Дата доступа: 04.03.2014.
6. Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2009. – 464 с.: ил. + CD-ROM

A. Kazinski, A. Gunosov The Relational Data Model for Software System for Calculation of the Total Load of the Department

The article is devoted to the description of data models for a software system for calculation of the total load of the Department of higher education institutions. Main stages and procedures of forming the total load of the Department. In the article the structure and peculiarities of implementation of the main elements of the developed model data. Describes some peculiarities of the transformation process of logical data model into a physical means MySQL Workbench. The relevance of the model is explained by the significant amount of work to calculate the total load of the Department and other tasks of automation of office management of the University. The efficiency of such work depends on the degree of use of modern information technologies.

Рукапіс паступіў у рэдакцыю 04.04.14