

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ГЕНЕРИРОВАНИЯ И ХРАНЕНИЯ РАБОЧИХ ПРОГРАММ

Введение. В современном мире актуально такое направление, как автоматизация различных процессов, а именно: составление и работа с какой-либо документацией и тому подобное. В качестве примера можно взять написание преподавателями ВУЗов рабочих программ учебных дисциплин. Данный документ содержит в себе информацию о дисциплине, на основании которой она будет преподаваться в дальнейшем. Объем рабочей программы от 10 до 20 страниц. При составлении преподавателем до пяти рабочих программ трудностей не возникает, так как выделенного на эту работу времени предостаточно. Но, если речь идет о большем количестве — то при работе могут появляться различные неточности и недочеты вследствие человеческого фактора. Эти проблемы впоследствии исправляются, однако на это тратится личное время преподавателя. Исходя из вышеперечисленного можно сделать вывод о том, что данный процесс можно упростить путем автоматизации. Проанализировав все возможные варианты и их плюсы, и минусы, было решено выбрать создание базы данных (далее — БД) генерирования, учета и хранения рабочих программ университета.

Основная часть. Первым этапом разработки было изучение объекта автоматизации — шаблона рабочей программы. В нем были проанализированы основные пункты, нюансы в зависимости от формы контроля, индикаторов и компетенций.

Вторым этапом было составление технического задания. Первым пунктом было написание общих сведений, которые включали в себя такие подпункты, как: объект автоматизации и документы, на основании которых создается система. Вторым было назначение и цели создания системы, а также были написаны критерии оценки достижения целей системы. Третий пункт — характеристика объектов автоматизации. В нем были описаны краткие сведения, в которых описывался процесс до автоматизации, и еще такой подпункт как сведения об условиях эксплуатации объекта автоматизации. Последний пункт — требования к системе. Были расписаны такие требования, как: общие, к видам обеспечения и к функциям, выполняемым системой.

Третий этап представляет из себя создание функциональной модели информационной системы с использованием методологии BPMN [1]. В ходе него разработались диаграммы подпроцессов функциональной модели трех уровней. Изначально создавалась модель верхнего уровня процессов, отображающая подпроцессы в информационной системе (рисунок 1).

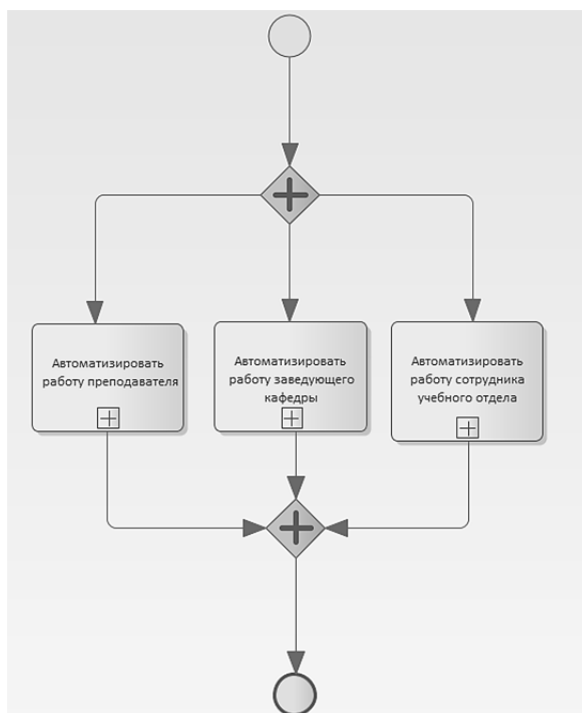


Рисунок 1 — Диаграмма подпроцессов функциональной модели

Затем проведлась декомпозиция подпроцессов, а именно: построение моделей, отображающих детали подпроцессов [2]. Это представляло из себя построение диаграмм декомпозиции подпроцесса «Автоматизировать работу преподавателя» (рисунок 2) и декомпозиции подпроцесса «Автоматизировать работу заведующего кафедры» (рисунок 3), которые в свою очередь представляют из себя второй уровень. Третий уровень — это построение диаграммы декомпозиции подпроцесса «Проверка и утверждение рабочих программ» (рисунок 4).

Последним этапом была разработка базы данных на основании всего выше сказанного. Для разработки была выбрана система управления реляционными базами данных, разработанная корпорацией Microsoft — Microsoft SQL Server [3][4].

В результате была разработана база данных, которая хранила основную информацию о учебных программах. В ней насчитывается 46 таблиц: AcademicDegrees, AcademicRanks, AudienceEducationalProgram, CompetenceLesson, Competences, DepartmentHeads, Departments, Disciplines, DisciplineTeacher, EducationalPrograms, EducationalProgramTrainingCourseForm, Faculties, InformationBlockContents, InformationBlocks, Inspectors, Lessons, Literatures, MethodicalRecommendations, FederalStateEducationalStandards, KnowledgeControlForms, MethodicalRecommendations, Protocols, Reviewers, Semesters, Specialties, Teachers и другие. В этих таблицах имеется Id, который используется как первичный ключ, который необходим для создания связей между таблицами. В данной базе данных имеются такие виды связей, как: один к одному, один ко многим, многие ко многим [5].

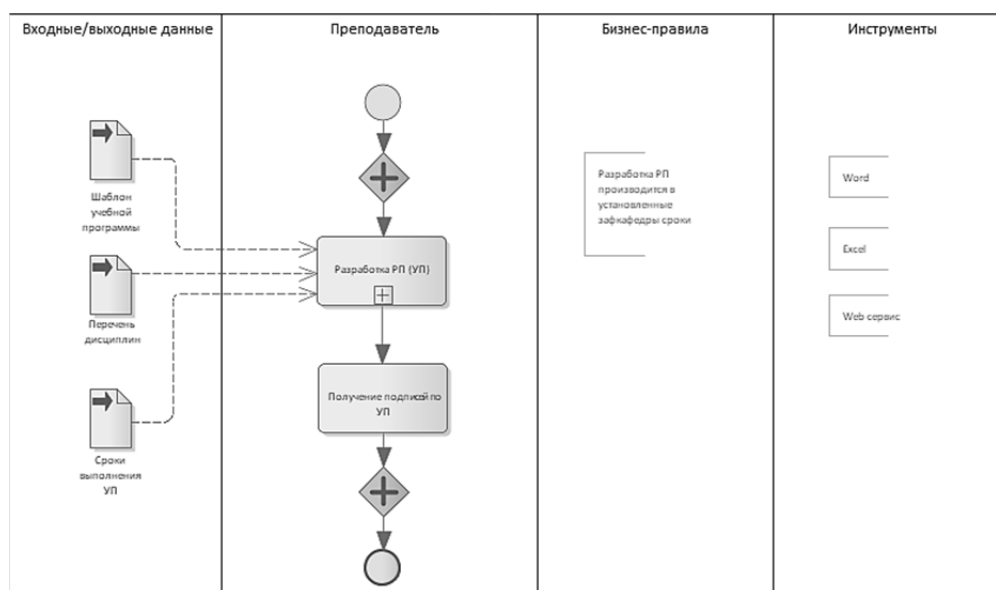


Рисунок 2 — Диаграмма декомпозиции подпроцесса «Автоматизировать работу преподавателя»

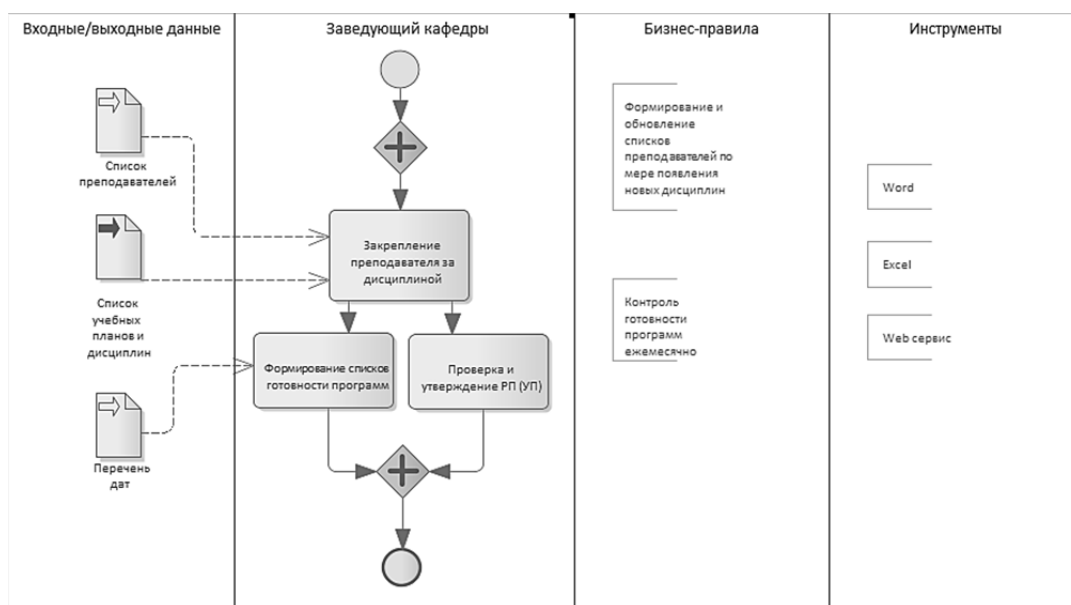


Рисунок 3 — Диаграмма декомпозиции подпроцесса «Автоматизировать работу заведующего кафедры»

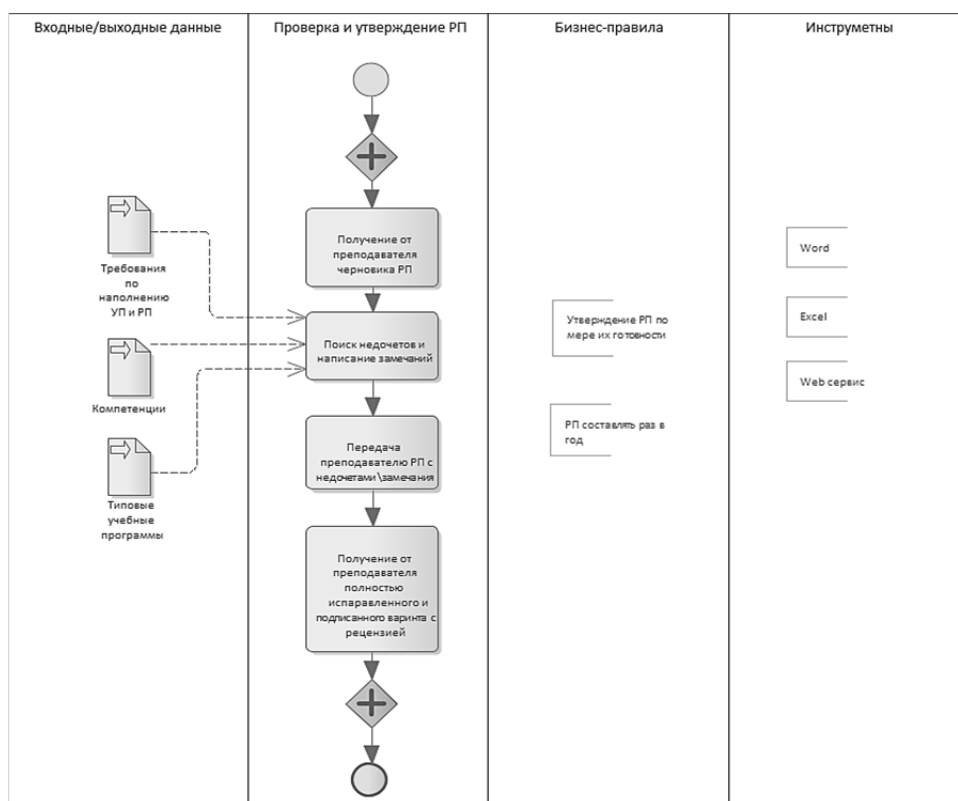


Рисунок 4 — Диаграмма декомпозиции подпроцесса «Проверка и утверждение рабочих программ»

Для реализации связи многие ко многим добавлены следующие таблицы: Lessons, Weeks, KnowledgeAssessment, Week, KnowledgeAssessment. Схема базы данных представлена на рисунке 5.

Описание основных таблиц БД:

- таблица “Disciplines”. В ней имеются такие колонки, как: название, её номер в учебном плане, трудоемкость в часах и в учебных единицах, контактная работа в часах, часы лекций, практических и лабораторных работ, семестр курсового проекта, курсовая работа и ее семестр, часы самостоятельной работы. А также имеются внешние ключи DepartmentId и CurriculumId;

- таблица “EducationalPrograms”. В ней имеются такие колонки, как: «дата утверждения», «дата рассмотрения» и «рекомендования к утверждению», «номер протокола», «ФИО главного библиотекаря», «статус» (не начата, в процессе, готова). Также имеются внешние ключи DisciplineId и ReviewerId;

- таблица “Departments”. В ней имеются такие колонки, как: «название кафедры», «сокращенное название» (заглавные буквы). Также имеется внешний ключ DepartmentHeadId;

- таблица “Faculties”. В ней имеется колонка «название»;

- таблица “Competences”. В ней имеются такие колонки, как: «код», «название»;

- таблица “Specialties”. В ней имеются такие колонки, как: «название специальности», «код», «название профиля», «квалификация», «форма и период обучения». А также имеются такие внешние ключи, как: DepartmentId, FederalStateEducationalStandardId и FacultyId;

- таблица “Indicators”. В ней имеются такие колонки, как: «номер», «описание». Также имеется внешний ключ CompetenceId.

Также для более удобной работы с БД были созданы такие хранимые процедуры, как:

1. AddDepartmentHead. Данная процедура предназначена для добавления заведующего кафедрой в базу данных.

2. AddDepartmentWithDepartmentHead. Выполняется после добавления заведующего кафедрой. Предназначена для добавления кафедры с закрепленным к ней заведующим.

3. AddDisciplines — предназначена для добавления дисциплины в БД.

4. AddSemesterDistribution — после выполнения процедуры добавления дисциплины выполняется данная процедура, которая закрепляет дисциплину за семестром.

5. AddTeacher — предназначена для добавления преподавателя.

6. GetAllDisciplinesByDepartment. Предназначена для получения информации о дисциплинах, которые закреплены за кафедрой по названию кафедры, а также дополнительной информации: кафедра, которая разрабатывает рабочую программу по данной дисциплине, специальность, статус готовности рабочей программы и фамилия преподавателя, который разрабатывает данную программу.

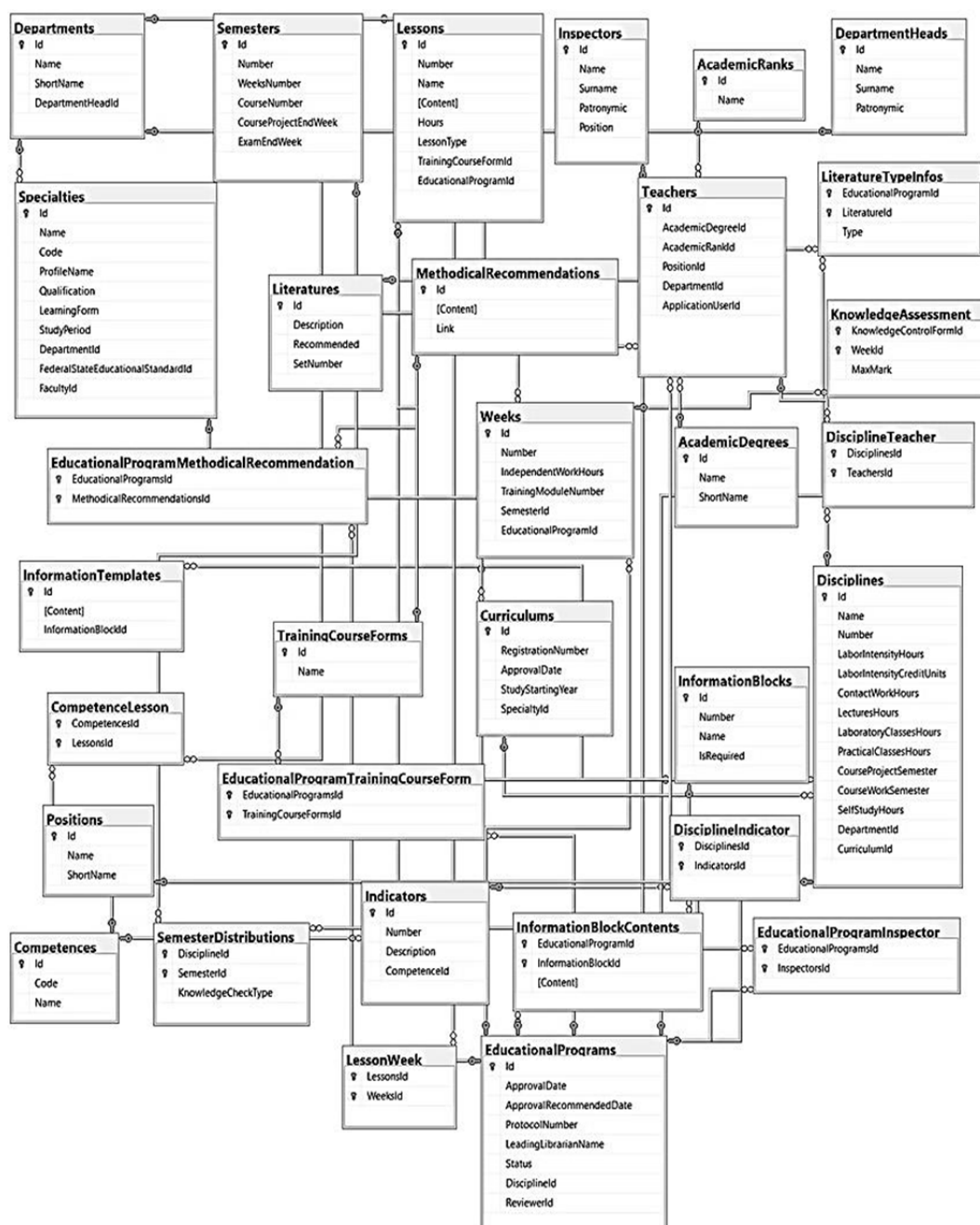


Рисунок 5 — Схема базы данных «Учет и хранения рабочих программ университета»

7. GetAllDisciplinesByDepartmentAndSpecialty. Получение информации о дисциплинах, которые закреплены за кафедрой для конкретной специальности. Поиск ведется по названию кафедры и специальности. Также выводится статус готовности рабочей программы, фамилия преподавателя, который разрабатывает рабочую программу, семестр и курс дисциплины.

8. GetCompetencesByDiscipline. Получение информации о компетенциях и индикаторах по названию дисциплины.

9. GetDisciplinesByCurriculumIdAndSemesterId. Получение информации о дисциплинах по id учебного плана и семестра.

10. GetSemesterDistributionByDisciplineId. Получение информации по id дисциплины о семестре, курсе и типе проверки знаний (экзамен, зачет).

11. GetTeachersByDisciplineId. Получение информации по id дисциплины о преподавателях, за которыми закреплена данная дисциплина, а именно: ФИО, ученое звание, должность и ученая степень.

12. GetTeachersDisciplines. Получение информации по фамилии преподавателя о дисциплинах, закрепленных за ним, и о статусе готовности рабочих программ по этим дисциплинам.

13. GetTeachersDisciplinesCount. Получение информации по названию кафедры о количестве рабочих программ, закрепленных за ней, а также фамилии преподавателей, которые разрабатывают рабочие программы.

Заключение. Разработанная база данных хорошо подходит для выполнения поставленной задачи, а именно автоматизация работы составления рабочей программы путем хранения основной информации в БД. Наличие хранимых процедур уменьшает загрузку сети между клиентами и сервером, а также упрощает работу по поиску различной часто необходимой информации. В дальнейшем рассматривается возможность использования различных ролей пользователей, курсоров, а также взятие данной БД для использования в разработке различных приложений, как вариант — приложение для генерирования шаблонов учебной программы.

Список цитируемых источников

1. *Стружкин, Н. П.* Базы данных : проектирование : учеб. для академ. бакалавриата / Н. П. Стружкин, В. В. Годин. — Люберцы : Юрайт, 2016. — 477 с.
2. *Кузнецов, С. Д.* Основы баз данных / С. Д. Кузнецов. — М. : Бином. Лаб. знаний, 2017. — 488 с.
3. *Петкович, Д.* Microsoft SQL Server 2008: рук. для начинающих / Д. Петкович. — М. : БХВ-Петербург, 2015. — 752 с.
4. *Тернстрем, Т.* Microsoft SQL Server 2008: разработка баз данных : учеб. курс Microsoft (+ CD-ROM) / Т. Тернстрем. — М. : Рус. ред., 2017. — 781 с.
5. *Фуфаев, Э. В.* Базы данных : учеб. пособие / Э. В. Фуфаев, Д. Э. Фуфаев. — М. : Академия, 2014. — 320 с