

ТРЕБОВАНИЯ К БАЗЕ ДАННЫХ В АВТОМАТИЧЕСКОМ РАСПИСАНИИ
УЧЕБНОГО ЗАВЕДЕНИЯ

Касимова Ш.Т.

Кафедра информационно-компьютерные технологии и программирование ТУИТ имени
Мухаммада ал-Хоразми, Ташкент, Республика Узбекистан

shoista.kasimova@mail.ru

<https://doi.org/10.5281/zenodo.7856362>

Аннотация. Современные требования, продиктованные реформированием экономики и общества, привели к значительному увеличению ресурсоемкости учебного процесса. Для снижения ресурсоемкости учебного процесса, обеспечения большей доступности обучения в образовательных учреждениях, обучающие технологии должны стать максимально эффективными, то есть обеспечивающими высокую степень экономичности учебного процесса при более высоком качестве обучения. Необходимо широкое применение инновационно-информационных методов обучения, интенсифицирующих учебный процесс. Всего этого можно достичь грамотным планированием учебного процесса в учебном заведении. В настоящей статье приведен один из способов решения проблемы в системах автоматизации составления расписания в ВУЗах

Ключевые слова: реляционные СУБД, автоматизация расписания, учебный план, база данных, динамический список.

ВВЕДЕНИЕ

В настоящее время основным предметом критики реляционных СУБД является некоторая ограниченность (прямое следствие простоты) при использовании в так называемых нетрадиционных областях, в которых требуются предельно сложные структуры данных. Еще одним часто отмечаемым недостатком реляционных баз данных является невозможность адекватного отражения семантики предметной области[1].

Любое приложение по составлению расписания в ВУЗе должно иметь свою базу данных, откуда она может считывать исходные данные, а также сохранять результаты алгоритма распределения занятий в расписании. Для этого база данных должна содержать ряд таблиц, информация из которых будет использоваться в процессе создания динамических списков различных объектов. Здесь понадобятся знания в применении объектно-ориентированного программирования, создании шаблонов и зависимостей между классами.

1. ОСНОВНАЯ ЧАСТЬ

Минимальным требованием к автоматическому составлению расписания является наличие списков групп, объединенных в потоки, преподавателей, аудиторий, подгрупп, дней и пар, типов занятий и учебного плана. Следовательно, предполагается наличие соответствующих таблиц.

Учитывая то, что в ТУИТ факультеты поочередно меняют смены обучения, была создана таблица, в которой имеются наименования факультетов и их смены. В таблице со списком потоков имеются id факультетов и соответствующие им потоки студентов. Эта таблица нужна для того, чтобы в процессе работы программы все поточные группы

находились на одной лекции, в одной аудитории, в одно и то же время (день, пара). Таблица «Группы» содержит потоки с соответствующими им группами.

В таблице с учебным планом предполагается наличие такой информации, как id занятий, группы, названия преподаваемых предметов, часы обучения, имена преподавателей, тип занятия, id подгруппы.

Стоит отметить, что наличие подгрупп в базе данных очень важно, т.к. иногда в учебном процессе встречается такая ситуация, что занятие той или иной группы подразумевает разделение группы на более мелкие подгруппы со своими преподавателями. А значит таблица «Подгруппы» должна содержать такие столбцы как группа, id подгруппы, предмет, преподаватель, тип урока. Под типом урока подразумевается лекция, практика, семинар или лабораторное занятие. Информационной системой эта таблица будет использована для того, чтобы занятия подгрупп ставились в одно и то же время, но (часто) в разные аудитории (обычно на занятиях по иностранным языкам). При этом должна происходить обязательная проверка на свободу всех преподавателей подгрупп в тот или иной момент времени [2].

В процессе работы программы этот список занятий преобразовывается в другую таблицу, в которой эти занятия будут дублироваться в зависимости от записанных часов. К примеру, в нашем университете 16 часов – это, когда занятие проходит один раз в две недели, 30 часов – каждую неделю, 46 часов – полтора раза в две недели, 60 часов – два раза в неделю и т.д. Такой расчет часов зависит от количества учебных недель. У нас их 15. Количество учебных недель должно храниться в таблице с информацией о составляемом расписании. К примеру, в этой таблице может также храниться такая информация как название составляемого расписания, учебный год, составитель и т.д.

Приведенная структура базы данных использовалась в работе алгоритма, который описывается следующим образом [3]:

1. Удаление старых результатов с базы данных. Здесь с помощью цикла удаляется каждая запись имеющегося расписания в базе.

2. Поиск подгрупп в таблице с учебным планом и автоматическая запись их в соответствующую таблицу с подгруппами. Это делается для того, чтобы пользователь информационной системы не тратил время на ручной ввод каждого занятия подгрупп всех групп.

3. Создание динамического списка с днями недели.

4. Создание динамических списков времени начала занятий. Так как в ТУИТ учеба протекает в две смены, создаются два именованных списка. При этом считается, что первая смена протекает с 1 по 4 пары, а вторая – с 3 по 6 пары.

5. Формирование списков аудиторий разных типов. Здесь создается 4 динамических списка с аудиториями: лекционными, практическими, лабораторными и спортзалами.

6. Формирование списков занятий.

7. Запуск распределения занятий в расписании. Запуск цикла, который будет распределять занятия в пространстве и времени, пока все занятия не будут готовы. В случае если невозможно покрыть все занятия по причине неграмотного ввода исходных данных в базу, цикл будет повторяться всего 7 раз.

Результирующая таблица содержит такие данные, как курс, на котором обучаются студенты той или иной группы занятия, саму группу, преподавателя, предмет, аудиторию, тип урока, номер пары, четность недели, день[4].

Любое приложение по составлению расписания в ВУЗе должно иметь свою базу данных, откуда она может считывать исходные данные, а также сохранять результаты алгоритма распределения занятий в расписании.

В процессе работы пришлось столкнуться с выбором между MySQL и SQLite.

SQLite — компактная встраиваемая реляционная база данных. Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором выполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала [6].

Несколько процессов или потоков могут одновременно без каких-либо проблем читать данные из одной базы. Запись в базу можно осуществить только в том случае, если никаких других запросов в данный момент не обслуживается; в противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

Именно вышеперечисленные трудности работы с SQLite стали причиной перехода на СУБД MySQL.

MySQL — свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы [7].

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Преимущества MySQL:

- Простота в работе — установить MySQL довольно просто. Дополнительные приложения, например, GUI, позволяет довольно легко работать с БД

- Богатый функционал — MySQL поддерживает большинство функционала SQL.
- Безопасность — большое количество функций, обеспечивающих безопасность, которые поддерживаются по умолчанию
- Масштабируемость — MySQL легко работает с большими объемами данных и легко масштабируется
- Скорость — упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

ЗАКЛЮЧЕНИЕ

Качество подготовки специалистов в вузах и особенно эффективность использования научно-педагогического потенциала зависят в определенной степени от уровня организации учебного процесса. Одна из основных составляющих этого процесса – расписание занятий – регламентирует трудовой ритм, влияет на творческую отдачу преподавателей, поэтому его можно рассматривать как фактор оптимизации использования ограниченных ресурсов – преподавательского состава и аудиторного фонда (7).

Разработкой алгоритмов по составлению графиков работы и расписаний занимается специальный раздел дискретной математики - теория расписаний. Задачи теории расписаний, разумеется, связаны с построением расписаний, т.е. с упорядочиванием некоторых работ (занятий, уроков) по времени и/или по исполнителям (преподавателям). При этом необходимо учитывать ограничения на последовательность выполнения работ, ограничения, связанные с исполнителями, и т.п. Цель решения таких задач – построение допустимых расписаний, при котором все ограничения соблюдены, или, что является более сложным, – нахождение оптимального допустимого расписания по тому или иному критерию оптимальности.

В результате проведенных исследований разработана программа автоматизации составления расписания ВУЗа, в которой учитывается информация об учебном плане, используемом в учебном процессе, название групп, объединяемых в потоки, название предметов, часы, выделяемые группе по тому или иному предмету, Ф.И.О преподавателей, типы занятий, названия всех факультетов и кафедр, списки аудиторий.

REFERENCES

1. Kasimova Sh.T., Kasimov S.R. Curriculum development as a task optimizing the educational process in educational institutions. International conference on information science and communication technologies ICIST 2019. Applications, Trends and Opportunities. DOI 10.1109/ICISCT47635.2019.9011930
2. Касимова. Ш.Т., Ахмаджанова С.Р. Об одном способе составления расписания занятий в учебном заведении. Воронеж : Издательство "Научно-исследовательские публикации", 2016. Материалы XVI Международной научно-методической конференции. стр. 392-396.
3. Семенов С.П., Татаринцев Я.Б. Сравнительный анализ подходов к автоматизации составления расписаний учебных занятий в образовательных учреждениях. Управление, вычислительная техника и информатика. Декабрь 2010 г., стр. 103-106.

4. Береговых Ю.В., Васильев Б.А, Володин Н.А. Алгоритм составления расписания занятий. г. Донецк, Украина : 2009 г., Искусственный интеллект, стр. 50-56.
5. Семенов С.П., Карминская Т.Д. Интегрированная информационная модель управления современным образовательным учреждением. *ИзвестияОрелГТУ*. Фундаментальные и прикладные проблемы техники и технологии: информационные системы и технологии, 1 Апрель 2008 г., №1–4/269, стр. 29-34.
6. T., Müller. Some Novel Approaches to Lecture Timetabling. *In Proceedings of the 4th Workshop of Constraint Programming for Decision and Control*. 1 Сентябрь 2002 г., стр. 31-37.
7. Касымова Ш.Т., Ахмаджанова С.Р. Об одной возможности адекватного отражения семантики в базах данных информационных систем. *Высшая школа*. Сентябрь 2016 г., стр. 39-40