

УДК 004.622

**ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ  
АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
КОМИССИИ ПО ДЕЛАМ НЕСОВЕРШЕННОЛЕТНИХ ПОЛОЦКОГО РАЙИСПОЛКОМА**

**К.В. МАРКЕВИЧ**

*(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*В статье рассматривается проектирование системы для работы с данными комиссии по делам несовершеннолетних. Проведен анализ технологий, наиболее подходящих, для разработки данной системы. Проведены исследования по актуальности разработки данной системы.*

**Введение.** Трудно представить хорошее клиент-серверное приложение без разработанной базы данных. Они позволяют удобно хранить большие объемы информации не на устройстве пользователя, а на удаленном сервере, который может эти данные предоставить по запросу от приложения.

Данная статья направлена на описание базы данных для разработанного приложения и методов ее проектирования.

**Основной раздел.** На данный момент, в Республике Беларусь, множество учреждений используют бумажный документооборот для хранения данных. Данный подход, с момента развития информационных технологий значительно уступил в эффективности, электронным системам хранения и обработки данных. В данном случае, электронные системы имеют ряд преимуществ:

- повышение качества управления административной деятельностью организации за счет комплексного внедрения автоматизированной информационной системы;
- сокращение времени оформления документации, посредством получения готовых данных из баз данных и компьютерного ввода;
- сокращение сроков предоставления отчетов и реестров услуг, на основе автоматически генерируемых шаблонов;
- оптимизация штата персонала учреждений

Данные критерии, несомненно доказывают эффективность данных систем, а также актуальность их разработки и внедрения в организацию.

**Средства решения задачи.** Основу разрабатываемой системы, составляет возможность хранить, накапливать и обрабатывать данные. Также в данном случае, данные должны храниться в строго систематизированном виде, основываясь, на нормах ведения документации, и имеют явную взаимосвязь.

Основываясь на этих данных необходимо будет использовать реляционную базу данных для хранения сущностей и язык запросов SQL для построения базы данных. В данном случае, использование именно реляционной базы данных, позволит формально и однозначно, определить ограничения целостности БД, в терминах ее концептуальной схемы.

**Достоинства реляционной модели.** Одним из важных достоинств реляционного подхода является его простота и доступность для понимания конечным пользователем. Единственной информационной конструкцией является таблица.

При проектировании реляционных баз данных применяются строгие правила, базирующиеся на математическом аппарате.

Реляционная модель обеспечивает полную независимость данных. При изменении структуры реляционной базы данных изменения, которые требуется произвести в прикладных программах, как правило, минимальны.

Манипулирование данными на уровне языка СУБД производится ненавигационно, поэтому для построения запросов и написания прикладных программ нет необходимости знания конкретной организации базы данных во внешней памяти. Конечно, при исполнении запросов на физическом уровне выполняется навигация по записям таблиц, однако эти действия производятся процедурами самой СУБД.

**Недостатки реляционной модели.** По сравнению с иерархической и сетевой моделями реляционная модель имеет более низкую скорость доступа и требует большего объема внешней памяти. В настоящее время этот фактор не является критическим вследствие многократно возросшего быстродействия компьютеров и такого же роста объема дисковой памяти.

Часто в результате логического проектирования появляется очень много таблиц, что затрудняет понимание структуры данных.

Далеко не всегда предметную область можно представить в виде совокупности таблиц. Так, в системах автоматизации проектирования и автоматизированной разработки программного обеспечения требуются гораздо более сложные структуры данных.

Для преодоления недостатков, присущих реляционной модели, в настоящее время развиваются постреляционная, многомерная и объектно-ориентированная модели. Эти модели в той или иной степени опираются на реляционную модель. Тем не менее реляционная модель и коммерческие продукты, основанные на этой модели, доминируют при построении экономических информационных систем.

Очевидно, что при реализации приложения важно сформировать некоторую архитектуру. В данном случае, для взаимодействия с базой данных, подойдет паттерн ADO.NET, представляющий собой взаимодействие 3 элементов: модель, представление и контроллер. Модель является некоторыми данными, которые описывают предметную область, и над ними могут быть выполнены некоторые операции. Также эта информация может быть предоставлена пользователю на графическом интерфейсе.

Приложение является этим самым графическим интерфейсом. Через данный интерфейс пользователи системы будут передавать данные на выполнение задачи контроллером. Контроллер, в свою очередь, реализует некоторые алгоритмы для получения данных с представления, обработки их и отправки обратно.

Цель моделирования данных состоит в обеспечении разработчика концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Для реализации приложения, необходимо установить все связи между сущностями: необходимо рассмотреть всю информационную систему в совокупности и определить взаимное влияние сущностей.

В соответствии выделенным сущностям, определен набор необходимых таблиц базы данных, представленный в таблице 1.

Таблица 1. – Перечень необходимых таблиц

Название таблицы	Описание предназначения
Parents	Таблица предназначена для хранения информации о родителях
Criminal	Таблица предназначена для хранения учетных данных о всех нарушителях
Offense	Таблица предназначена для хранения специфических данных

Важное достоинство баз данных – возможность хранить сгруппированные данные в разных таблицах и устанавливать связи между этими таблицами посредством ключей и последующее объединение этих данных в единую базу. Это позволяет снизить избыточность данных, которые хранятся, упростить организацию запросов для их получения.

Первичный ключ служит ограничителем целостности в рамках одной таблицы для идентификации, т.к. поле первичного ключа не может повторяться или быть пустым. Важное условие - каждая таблица может содержать только один первичный ключ.

Для организации связей между двумя и более таблицами, используются вторичные ключи. Они служат ограничителями целостности связей нескольких таблиц, т.к. подчиненная таблица не может ссылаться на несуществующие записи главной таблицы (что позволяет строить целостные модели данных).

Существует три вида связей между таблицами:

- «один-ко-многим»,
- «один-к-одному»,
- «многие-ко-многим».

Связь «один-ко-многим» - самая распространенная для реляционных баз данных. Для обеспечения ссылочной целостности в дочерней таблице создается внешний ключ, через который организовывается связь с дочерней таблицей. Обязательное условие состоит в том, что внешний ключ по составу полей должен совпадать с первичным ключом родительской таблицы.

**Заключение.** В ходе данного исследования была спроектирована, программная система, наиболее подходящая по архитектурным особенностям под основу для разработки приложения для работы с данными. При этом разработанная архитектура, является универсальной для любого современного языка программирования, а также оставляет возможности для реализации необходимого функционала.