# ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

На сегодняшний день школы, институты и другие учебные учреждения не предоставляют ученикам и студентам достаточного экономического образования, и вследствие чего, большинство выходят во взрослую жизнь финансово безграмотными.

Чтобы не оказаться на грани банкротства в начала карьеры или в ее разгаре, необходимо иметь кое-какие привычки. По мнению экспертов, одной из важнейших привычек — это вести ежедневный учет расходов. Ежедневный учет расходов также необходим каждому из нас, как и ежедневные процедуры ухода за лицом, телом и волосами.

Ежедневные расходы можно вести в обыкновенном блокноте, ежедневнике, однако в этом случае, придется составлять отчеты самостоятельно, а также блокнот можно потерять или забыть. Именно поэтому было принято решение, перенести ежедневный учет расходов в электронный формат.

Основной задачей курсового проекта является автоматизация ведения учета расходов и отчета по ним.

В качестве основных запросов в БД можно выделить следующие запросы:

* CRD операции с расходами
* Чтение выборки расходов за определённый срок (год, месяц, день)
* Отчёт по расходам – сумма расходов с одинаковой категорией за определённый срок

# ВЫБОР СРЕДСТВ И МЕТОДОЛОГИЙ ПРОЕКТИРОВАНИЯ. ВЫБОР СУБД

В качестве основной СУБД была выбрана SQLite. Легко встраивается в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам, вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

Преимущества SQLite:

* Файловая структура — вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины
* Используемые стандарты — хотя, может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), но основные все - таки поддерживаются
* Отличная при разработке и тестировании — в процессе разработки приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки, написанной на языке C.

Несомненно, данную СУБД стоит использовать во встроенных приложениях — если нет необходимости в простом переносе приложения и не важна масштабируемость. Например, однопользовательские приложения, мобильные приложения или игры, но для данного курсового проекта она подходит, так как нет необходимости вносить сотни тысяч записей одновременно с разных устройств. Однако при масштабируемости, возможен быстрый переход на PostgreSQL из - за использования Django ORM.

Django ORM позволяет связывать базы данных с концепциями объектно-ориентированных языков программирования, таких как Python, создавая «виртуальную объектную базу данных».

# ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ

Для реализации задуманного проекта, необходимо построить концептуальную модель базы данных, так как качественно спроектированная база данных необходима приложению так же, как и прочный фундамент необходим дому.

Сперва необходимо построить модель, которая свяжет таблицу «Расходы» и таблицу «Категория». Они будут находиться в связи 1 к 1, так, что один расход может иметь только одну категорию. Модель изображена на рисунке 1.

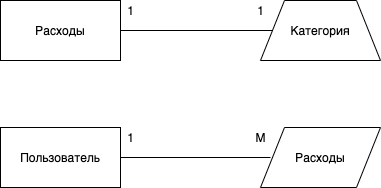


Рисунок 1 – Модель «Расходы» - «Категория»

Для того, чтобы связать таблицы «Пользователь» и «Расходы», потребуется связь 1 ко многим, потому что у одного пользователя может быть много расходов. Модель изображена на рисунке 2.

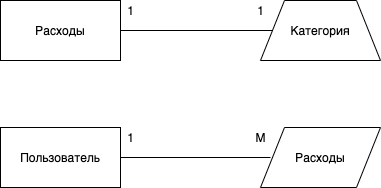


Рисунок 2 – Модель «Пользователь» - «Расходы»

# ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ БД

Для создания качественной базы данных, необходимо спроектировать ее логическую структуру.

Для начала определяются требуемые поля:

* Email
* Имя
* Пароль
* Баланс
* Сумма расхода
* Категория расхода
* Дата расхода
* Иконка расхода

Далее необходимо разделить поля на таблицы и определить типы и размеры полей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Пользователь | |  | Расходы | |  |
| Email | varchar(254) |  | Сумма | integer |  |
| Имя | varchar(150) |  | Категория | Integer(FK) |  |
| Пароль | varchar(128) |  | Дата | date |  |
| Баланс | integer |  | Пользователь | Integer(FK) |  |
|  |  |  |  |  |  |
| Категория |  |  |  |  |  |
| Название | varchar(100) |  |  |  |  |
| Иконка | varchar(100) |  |  |  |  |

Следующим этапом будет построение базовой модели и реализация её в СУБД. Базовая модель представлена на рисунке 3.



Рисунок 3 – Базовая модель в СУБД