**3 ПРОЕКТИРОВАНИЕ**

**3.1 Разработка архитектуры программного продукта**

Проектирование программного обеспечения — процесс создания проекта программного обеспечения (ПО), а также дисциплина, изучающая методы проектирования.[5] Проектирование ПО является частным случаем проектирования продуктов и процессов.

Целью проектирования является определение внутренних свойств системы и детализации её внешних (видимых) свойств на основе выданных заказчиком требований к ПО (исходные условия задачи). Эти требования подвергаются анализу.

Проектирование ПО включает следующие основные виды деятельности:

• выбор метода и стратегии решения;

• выбор представления внутренних данных;

• разработка основного алгоритма;

• документирование ПО;

• тестирование и подбор тестов;

• выбор представления входных данных.

Первоначально программа рассматривается как чёрный ящик. Ход процесса проектирования и его результаты зависят не только от состава требований, но и выбранной модели процесса, опыта проектировщика.

Модель предметной области накладывает ограничения на бизнес-логику и структуры данных.

В зависимости от класса создаваемого ПО, процесс проектирования может обеспечиваться как «ручным» проектированием, так и различными средствами его автоматизации. В процессе проектирования ПО для выражения его характеристик используются различные нотации — блок-схемы, ER-диаграммы, UML-диаграммы, DFD-диаграммы, а также макеты. Развитие современных информационных технологий приводит к возрастанию сложности информационных систем. Методологии, технологии и инструментальные средства проектирования (CASE-средства) составляют основу проекта любой информационной системы.

Современные CASE-средства охватывают обширную область поддержки многочисленных технологий проектирования ИС: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл ПО.

Наиболее трудоемкими этапами разработки ИС являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение структурных или иных диаграмм в реальном масштабе времени. Графические средства моделирования предметной области позволяют разработчикам в наглядном виде анализировать существующую ИС, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями.

При построении информационной модели применялся стандарт IDEF1x. Модель IDEF1x создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения как части документации, необходимой в цикле разработки. Диаграмма IDEF1x строится из трех основных элементов - сущностей, атрибутов и связей. Если рассматривать диаграмму как графическое представление правил предметной области, то сущности являются существительными, а связи — глаголами.

Сущность - логическое понятие. Сущности соответствует таблица в реальной СУБД. В IDEF1x сущность визуально представляет 3 основных вида информации:

– атрибуты, составляющие первичный ключ. Для каждого первичного ключа IDEF1x создает при генерации структуры БД уникальный индекс;

– не ключевые атрибуты;

– тип сущности (независимая/зависимая).

Связь — это функциональная зависимость между двумя сущностями (в частности, возможна связь сущности с самой собой). Связь - это понятие логического уровня, которому соответствует внешний ключ на физическом уровне. В IDEF1x связи представлены 5 основными элементами информации:

– связи;

– родительская сущность;

– дочерняя (зависимая) сущность;

– мощность связи;

– допустимость пустых значений.

Связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируются через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой.

Целевая СУБД, имена объектов и тины данных, индексы составляют второй (физический уровень модели IDEF1x).

Процесс построения информационной модели состоит из следующих шагов:

– определение сущностей;

– определение зависимостей между сущностями;

– задание первичных и альтернативных ключей;

– определение атрибутов сущностей;

– приведение модели к требуемому уровню нормальной формы;

– переход к физическому описанию модели - назначение соответствий: имя сущности — имя таблицы, атрибут сущности — атрибут таблицы; задание триггеров, процедур и ограничений;

– генерация базы данных.

В ходе построения модели было предложено следующее решение (см. рисунок 2):

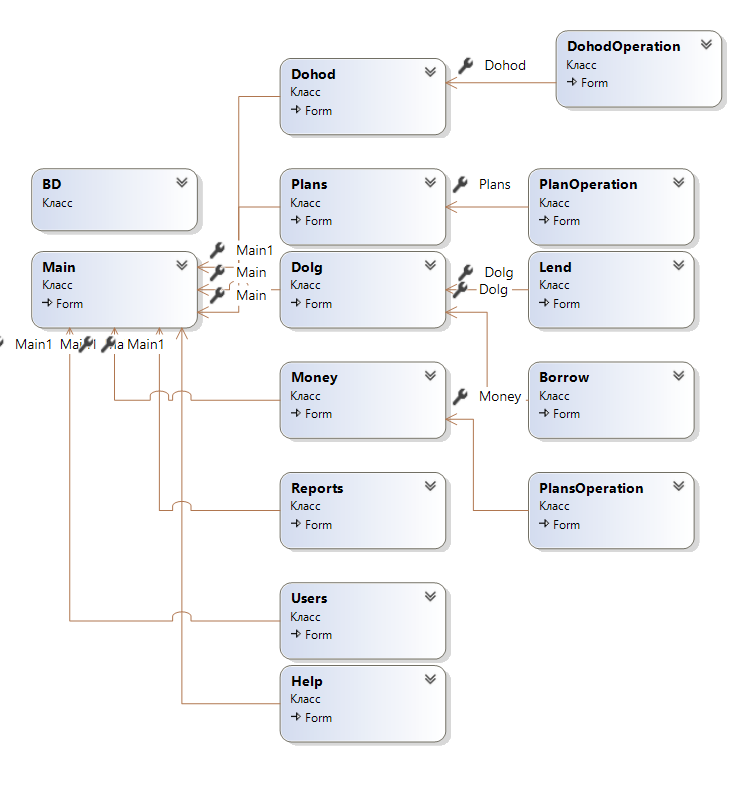


Рисунок 2 - Информационная модель системы

## Проектирование структур хранения данных

Чтобы построить схему реляционной базы данных необходимо определить совокупность отношений, которые составляют базу данных. Эта совокупность отношений будет содержать всю информацию, которая должна храниться в базе данных.

На рисунке 3.1 представлены отношения для базы данных организации по прокату автомобилей.

В реляционной базе данных каждому объекту и сущности реального мира соответствуют кортежи отношений. И любое отношение должно обладать первичным ключом. Ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждое отношение должно обладать хотя бы одним ключом.[6] В таблице 2.1 определены первичные ключи для отношений.

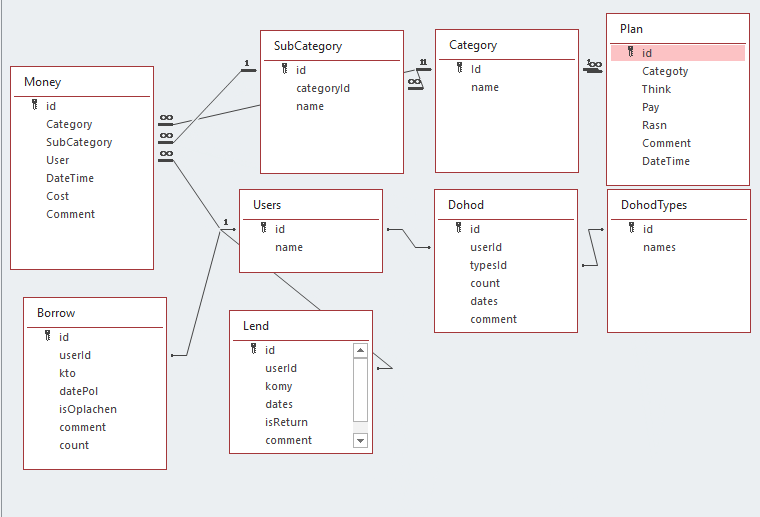


Рисунок 3.1– Набор необходимых отношений базы данных

Процесс преобразования базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную избыточность, то есть нормализация не имеет целью уменьшение или увеличение производительности работы или же уменьшение, или увеличение объёма БД. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в БД информации.

Для реляционных баз данных необходимо, чтобы все отношения базы данных обязательно находились в 1НФ. Нормальные формы более высокого порядка могут использоваться разработчиками по своему усмотрению. Однако грамотный специалист стремится к тому, чтобы довести уровень нормализации базы данных хотя бы до 3НФ, тем самым, исключив из базы данных избыточность и аномалии обновления.

Ограничение целостности отношений заключается в том, что в любом отношении должны отсутствовать записи с одним и тем же значением первичного ключа. Конкретно требование состоит в том, что любая запись любого отношения должна быть отличной от любой записи другой записи этого отношения. Это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

У проектируемой все таблицы будут иметь первичные ключи:

На основе результатов выполненных действий была сформирована схема реляционной базы данных. На этой схеме отображены все отношения базы данных, а также связи между внешними и первичными ключами.

Таким образом, были определены ключевые атрибуты, связи между отношениями, организована целостность данных, нормализация некоторых отношений. Итоговым моментом стало построение нормализованной схемы разрабатываемой базы данных.

В создаваемой базе данных будут использоваться следующие типы данных:

1. char [(n)] - строковое значение постоянной длины;
2. date - календарная дата (год, месяц, день);
3. numeric [(p)] - точное численное значение;
4. text - строковое значение переменной длины;