

## РАЗДЕЛ 1. ОСНОВЫ МЕТОДОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### **Цель:**

познакомиться с принципами нормализации модели данных «сущность-связь».

### **Задачи:**

- изучить правила нормальных форм;
- научиться использовать CASE-дизайнер для построения модели данных;
- овладеть приемами работы в CASE-инструментарии.

## ЛАБОРАТОРНАЯ РАБОТА 2. РАЗРАБОТКА МОДЕЛИ В CASE-ДИЗАЙНЕРЕ, ГЕНЕРАЦИЯ ФИЗИЧЕСКОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ

### Нормализация данных

Нормализация данных – это процесс приведения модели к виду, позволяющему получить в дальнейшем структуру базы данных, в которой устранена избыточность хранения и сведены к минимуму аномалии при добавлении, удалении, изменении данных. В процессе нормализации модель должна быть последовательно приведена к первой, второй, третьей нормальным формам (1НФ, 2НФ, 3НФ).

Приведение к 1НФ состоит в исключении множественных или повторяющихся атрибутов.

Приведение к 1НФ осуществляется по следующим правилам:

- множественный атрибут или группа таких атрибутов выделяются в самостоятельную сущность;
- между исходной и новой сущностями устанавливается связь.

Первая нормальная форма, таким образом, является механизмом выявления упущенных сущностей и связей.

Приведение ко 2НФ предполагает вынесение атрибутов, которые не удовлетворяют требованиям функционально полной зависимости от уникального идентификатора сущности, являющегося составным ключом.

Атрибут **A** функционально зависит от атрибута **B** ( $\Phi 3 : B \rightarrow A$ ), если каждому значению **A** соответствует не более одного значения **B**. Функциональная зависимость может иметь место не только от одного атрибута, но и от группы атрибутов –  $\Phi 3 : B, C, D \rightarrow A$ .

Функциональная зависимость является полной, если атрибут зависит от составного ключа и не зависит ни от какого подмножества данного ключа, т. е. имеет место зависимость  $\Phi 3 : K1, K2, \dots, Kn \rightarrow A$  и не справедлива ни одна из следующих зависимостей:  $\Phi 3 : K1 \rightarrow A$ ;  $\Phi 3 : K2 \rightarrow A$ ; ...  $\Phi 3 : K1, K2 \rightarrow A$  и т. д.

Приведение к 2НФ осуществляется по следующим правилам:

- атрибуты, зависящие от части составного ключа, и сама эта часть выносятся в отдельную (новую) сущность и исключаются из исходной;
- ключом новой сущности становится часть ключа исходной, от которой имеет место функционально неполная зависимость;
- между исходной и новой сущностями устанавливается связь типа  $M : 1$ , ключевая со стороны «многие».

Приведение к 2НФ, так же как и приведение к 1НФ, позволяет выявить в модели упущенные сущности и связи и, кроме того, устранить избыточность данных в исходной модели.

Приведение к 3НФ состоит в исключении транзитивных зависимостей атрибутов от атрибутов, не являющихся частью ключа.

Атрибут **A** транзитивно зависит от **C**, если **A** зависит от **B**, а **B** зависит от **C** ( $\Phi 3 : C \rightarrow^* A$ , если  $\Phi 3 : B \rightarrow A$  и  $\Phi 3 : C \rightarrow B$ ).

Приведение к 3НФ осуществляется по следующим правилам:

- атрибуты, зависящие от неключевых атрибутов, и сами эти атрибуты выносятся в отдельную (новую) сущность и исключаются из исходной сущности;
- ключом новой сущности становятся те атрибуты исходной, от которых была обнаружена транзитивная зависимость;
- между исходной и новой сущностями устанавливается связь типа  $M : 1$ .

Приведение к 3НФ, так же как и приведение к 1НФ, 2НФ, позволяет выявить в модели упущенные сущности и связи и, кроме того, устранить избыточность данных и аномалии изменения.

## Проектирование структуры базы данных

На основе модели «сущность-связь» может быть автоматически синтезирована структура базы данных. Генерация осуществляется в соответствии со следующими правилами:

- каждая сущность преобразуется в таблицу, имя сущности становится именем таблицы;
- атрибуты сущности преобразуются в колонки таблицы, имена атрибутов становятся именами колонок;
- ключевые атрибуты становятся первичными ключами таблицы;
- если для сущности была определена ключевая связь, то первичный ключ таблицы для связанной сущности копируется и объединяется с ключом таблицы для рассматриваемой сущности;
- связи  $M : 1$  и  $1 : 1$  приводят к копированию первичных ключей таблицы для сущности, находящейся на одной стороне связи, в таблицу для сущности, находящейся на другом конце связи, если связь  $M : 1$ , то ключи таблицы для

сущности, находящейся на конце «один» копируются в таблицу для сущности, находящейся на конце «многие».

#### Задание для самостоятельной работы

1. Разработайте концептуальную модель вашего варианта в CASE – редакторе PowerDesigner (рис.1).
2. Выполните проверку модели Tools – Check Model.
3. В случае успешной проверки запустите процесс создания физической структуры базы данных: Tools – Generate Physical Data Model.
4. Создайте сценарий для формирования таблиц Database – Generate Database.
5. После создания сценария должно появиться диалоговое окно Generated Files, в котором можно перейти в режим просмотра сценария в Блокноте с помощью кнопки Edit.
6. Скопируйте в буфер обмена текст сценария.
7. Откройте Microsoft SQL Management. Создайте новую базу данных.
8. Перейдите в режим Query, скопируйте сценарий из буфера обмена в окно редактора запросов и запустите сценарий на исполнение (рис.2).

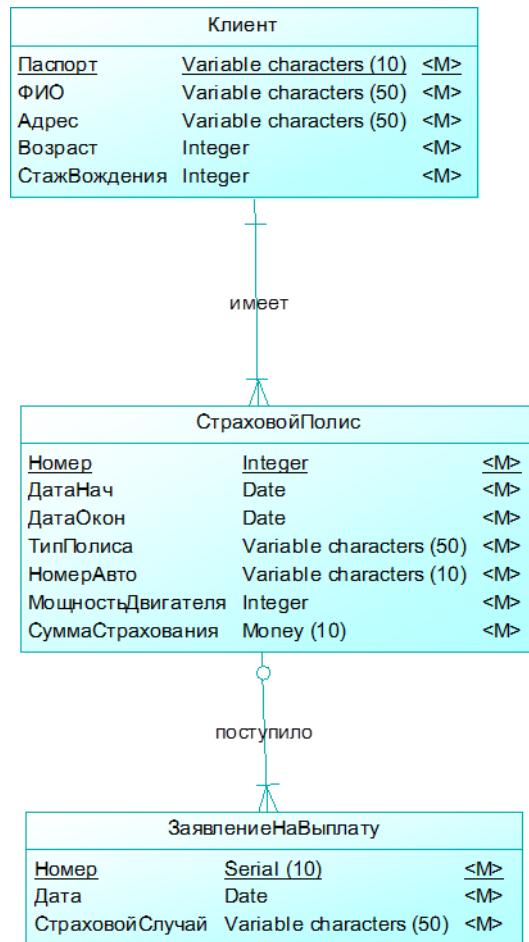


Рис.1. Пример концептуальной модели для Страхования автогражданской ответственности

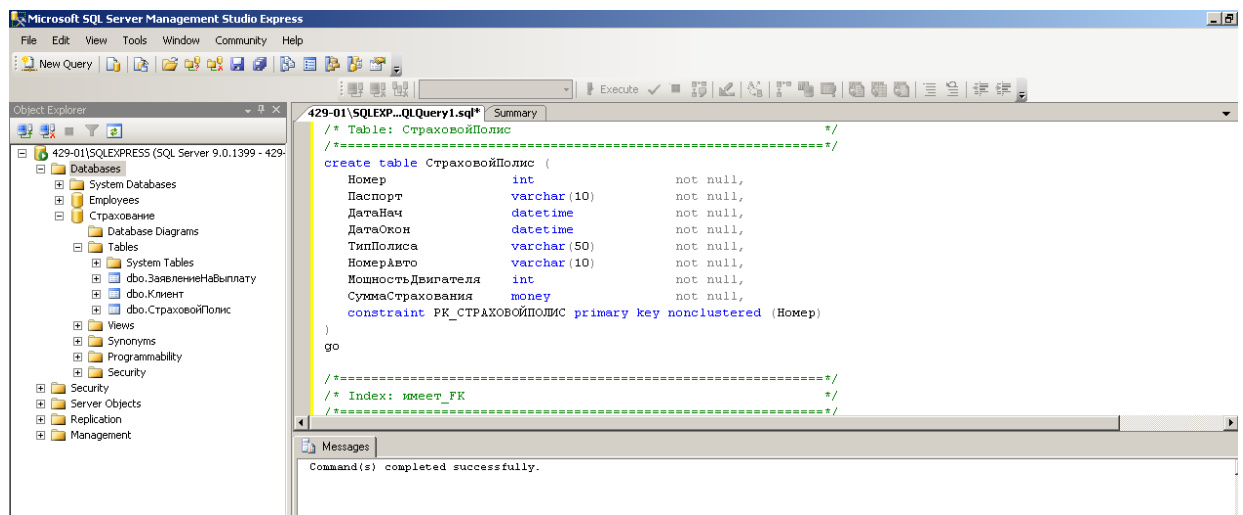


Рис.2. Выполнение сценария для создания таблиц базы данных

Вернитесь в окно редактора Microsoft SQL Management. Добавьте в структуру базы данных диаграмму (рис.3).

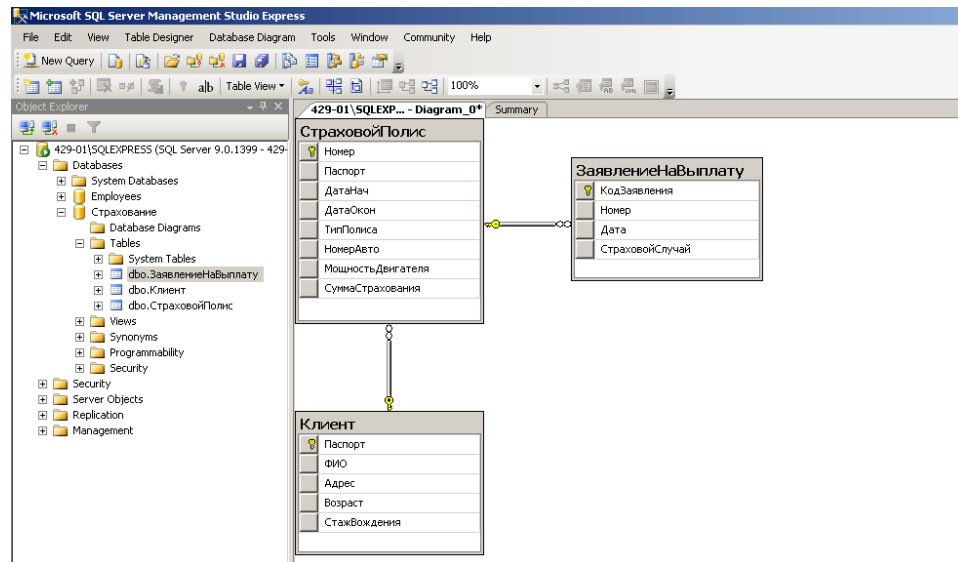


Рис.3. Диаграмма таблиц и связей

### Содержание отчета

1. Титульный лист
2. Скрипт создания базы данных
3. Копия экрана из SQL server с диаграммой таблиц и связей.