

УДК 004.021

## ПРИНЦИПЫ МОДЕЛИРОВАНИЯ БАЗЫ ДАННЫХ НА ПРИМЕРЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ТОВАРОВ

М.А. ТРАЩЕНКО

(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЁНОК)

*Рассматриваются основные принципы моделирования базы данных на примере автоматизированной информационной системы под операционную систему Windows учета товаров для продвижения малого бизнеса компании «GrandDecor».*

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

В представленной статье опишем методы проектирования базы данных на примере автоматизированной информационной системы (АИС) под операционную систему Windows учета товаров для продвижения малого бизнеса компании «GrandDecor».

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 любой системе баз данных.

Сущность структурного подхода к разработке АИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. Общая модель системы строится в виде некоторой иерархической структуры, которая отражает различные уровни абстракции с ограниченным числом компонентов на каждом из уровней. Одним из главных принципов структурного системного анализа является выделение на каждом из уровней абстракции только наиболее существенных компонентов или элементов системы [1].

В структурном анализе используется ряд средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм). Наиболее распространенной диаграммой, предназначенной для моделирования данных, является диаграмма Entity-Relationship Diagrams (ERD) «сущность-связь».

В результате краткого анализа предметной области выявлен необходимый набор сущностей (таблиц), представленных в таблице.

Таблица. – Список всех таблиц, образующих базу данных

Наименование таблицы	Назначение
Брак	Хранит информацию о товарах, на которых выявлен брак
Категория товара	Хранит информацию о категориях товара, реализуемых на торговых точках
Номенклатура товара	Хранит название, идентификационный номер и категорию товара
Позиция ТТН	Хранит информацию о товаре, оформленном с номером данной ТТН
Поставщики	Хранит информацию о поставщиках товаров
Продажа	Хранит информацию о продажах товара
Производитель	Хранит условно-постоянную информацию (Справочник подстатей расходов)
Склад	Хранит информацию о производителях товара, поставляемого поставщиками
Тип поставщика	Хранит информацию о категории товара, который поставляет данный поставщик
Товар	Хранит полную информацию о товаре
ТТН	Хранит информацию об оформленной ТТН

Базы данных могут содержать таблицы, которые связаны между собой различными связями. Связь (relationship) представляет ассоциацию между сущностями разных типов. При выделении связи выделяют главную или родительскую таблицу (primary key table / master table) и зависимую, дочернюю таблицу (foreign key table / child table). Дочерняя таблица зависит от родительской. Для организации связи используются внешние ключи. Внешний ключ представляет один или несколько столбцов из одной таблицы, который одновременно является потенциальным ключом из другой таблицы. Внешний ключ необязательно должен соответствовать первичному ключу из главной таблицы. Хотя, как правило, внешний ключ из зависимой таблицы указывает на первичный ключ из главной таблицы.

Реляционная схема базы данных представлена на рисунке.

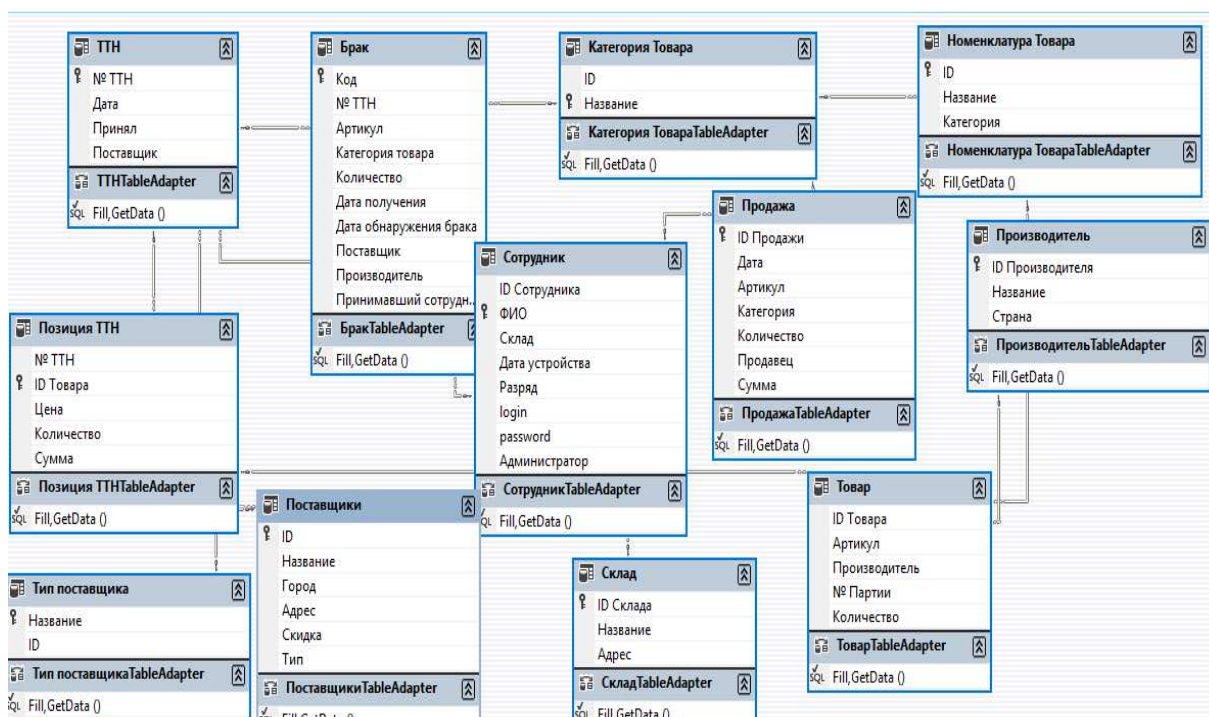


Рисунок. – Реляционная схема базы данных

Реляционная модель представляет базу данных в виде множества взаимосвязанных отношений (таблиц), которые используются для хранения информации об объектах, представленных в базе 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).

Связи между таблицами бывают следующих типов:

1. Один к одному (One to one);
2. Один к многим (One to many);
3. Многие ко многим (Many to many).

*Связь один к одному.* Данный тип связи встречается не часто. В этом случае объекту одной сущности можно сопоставить только один объект другой сущности. Например, на некоторых сайтах пользователь может иметь только один блог. То есть возникает отношение один пользователь – один блог.

*Связь один ко многим.* Это наиболее часто встречаемый тип связи. В этом типе связи несколько строк, из дочерней таблицы, зависят от одной строки в родительской таблице. Например, в одном блоге

может быть несколько статей. В этом случае таблица блогов является родительской, а таблица статей – дочерней. То есть один блог – много статей.

*Связь многие ко многим.* При этом типе связей одна строка из таблицы А может быть связана с множеством строк из таблицы В. В свою очередь одна строка из таблицы В может быть связана с множеством строк из таблицы А. Типичный пример – студенты и курсы: один студент может посещать несколько курсов, и соответственно на один курс могут записаться несколько студентов [2].

Таким образом, рассмотренные варианты проектирования, несомненно, являются актуальными и, при верном подходе к выбору, позволяют произвести проектирование базы данных.

#### ЛИТЕРАТУРА

1. Helpiks [Электронный ресурс] // Моделирование данных. – Режим доступа: <https://helpiks.org/6-43047.html>. – Дата доступа: 13.09.17.
2. Metanit [Электронный ресурс] // Реляционные базы данных. – Режим доступа: <https://metanit.com/sql/tutorial/1.3.php>. – Дата доступа: 14.09.18.