

УДК 681.3

В. А. Игнатюк<sup>1</sup>, Е. А. Сторожок<sup>2</sup>

## ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ НА ОСНОВЕ ПРИНЦИПОВ НОРМАЛИЗАЦИИ

---

---

*В рамках данной работы проводится исследование возможностей автоматизации процесса нормализации схемы реляционной базы данных. Важным этапом проектирования реляционной БД является логическое проектирование. Данный этап выполняется после выбора конкретной СУБД. Чаще всего логическое проектирование базируется на принципах нормализации. Результат логического проектирования – схема БД (совокупность схем отношений, входящих в БД). Принцип нормализации предполагает удаление из отношений БД избыточных зависимостей, которые могут привести к нарушению целостности данных в ходе выполнения транзакций. Процесс нормализации даёт несколько вариантов третьей нормальной формы. Возникает необходимость выбора из этих вариантов наилучшего. Данный выбор можно организовать с использованием таких критериев, как объём занимаемой памяти и скорость выполнения транзакций.*

**Ключевые слова:** База данных, нормализация, логическое проектирование, отношение, схема отношения, функциональная зависимость, ключевой атрибут, нормальная форма, транзитивная зависимость, алгоритм.

Лидирующее положение среди автоматизированных информационных систем занимают системы, реализованные на основе реляционной базы данных (БД). Важным этапом проектирования реляционной БД является логическое проектирование. Данный этап выполняется после выбора конкретной СУБД. Чаще всего логическое проектирование базируется на принципах нормализации. Результат логического проектирования – схема БД (совокупность схем отношений, входящих в БД). Принцип нормализации предполагает удаление из отношений БД избыточных зависимостей, которые могут привести к нарушению целостности данных в ходе выполнения транзакций. Процесс нормализации даёт несколько вариантов третьей нормальной формы. Возникает необходимость выбора из этих вариантов наилучшего. Данный выбор можно организовать с ис-

---

<sup>1</sup> © Виктор Александрович Игнатюк, д-р физ.-мат. наук, профессор кафедры электроники Владивостокского государственного университета экономики и сервиса, ул. Гоголя д. 41.

<sup>2</sup> © Евгений Анатольевич Сторожок, канд. техн. наук, доцент кафедры информационных технологий ТОВМИ им. С.О. Макарова, пер. Камский д. 6, E-mail storea@mail.ru.

пользованием таких критериев, как объём занимаемой памяти и скорость выполнения транзакций.

Фундаментальным понятием нормализации является понятие функциональной зависимости (ФЗ). Функциональная зависимость – такая зависимость между атрибутами отношения, когда в каждый момент времени одному значению атрибута  $X$  соответствует только одно значение атрибута  $Y$ . ФЗ записывается:  $X \rightarrow Y$ .

С целью устранения избыточных зависимостей производится декомпозиция исходных отношений. Различают пять нормальных форм. На практике достаточным бывает приведение отношения к третьей нормальной форме (3 НФ). Условием нахождения отношения в 1 НФ является атомарность значений атрибутов, т.е. на пересечении каждой строки и каждого столбца находится ровно одно значение, а не набор значений. Отношение находится в 2 НФ, если оно находится в 1 НФ и все неключевые атрибуты функционально полно зависят от составного ключа. Полная функциональная зависимость подразумевает, что неключевой атрибут зависит от всего составного ключа и не зависит от отдельных элементов этого ключа. Отношение находится в 3 НФ, если оно находится в 2 НФ и в нём отсутствуют транзитивные зависимости. Считается, что атрибут  $Z$  транзитивно зависит от атрибута  $X$ , если существуют ФЗ:  $X \rightarrow Y, Y \rightarrow Z$ .

Обратные же зависимости отсутствуют, т.е.  $Y \not\rightarrow X, Z \not\rightarrow Y$ .

Существующие СУБД не предусматривают автоматизацию процесса нормализации, следовательно, решаемая в данной работе задача актуальна.

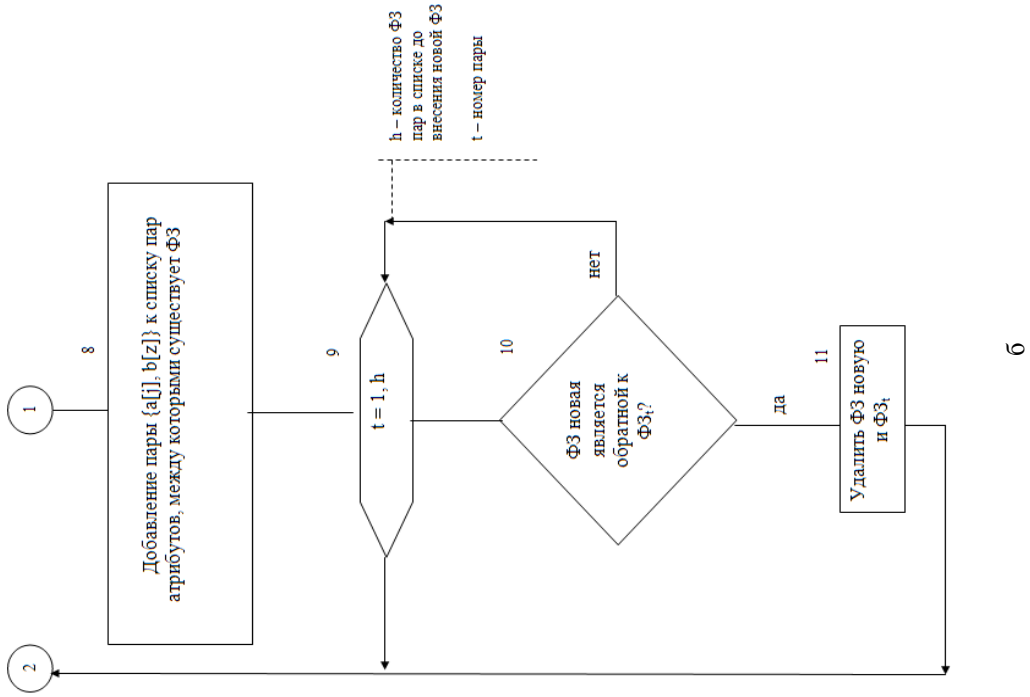
В рамках данной работы проводятся исследования возможностей автоматизации процесса нормализации. Предположим, что в исходной БД все отношения находятся в 1 НФ.

Приведение ко второй нормальной форме лучше всего реализуется путём добавления нового атрибута, называемого «идентификационный номер» (ID), который становится новым ключом. Таким образом, составной ключ меняется на простой и частичные ФЗ не могут возникнуть.

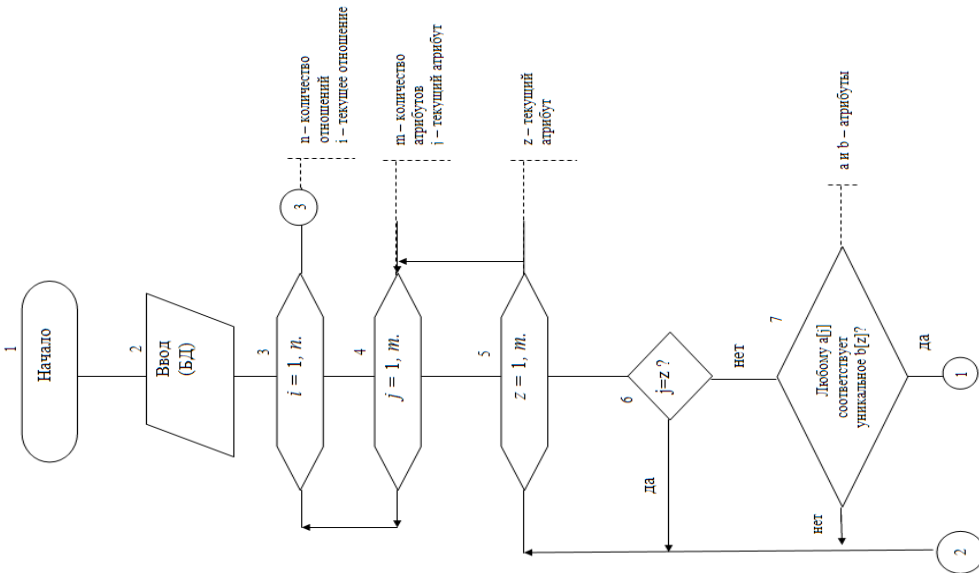
Проведенные в ходе работы исследования показали, что наличие избыточных зависимостей может привести к нарушению целостности данных при выполнении транзакций. Избыточные зависимости устраняются в процессе логического проектирования БД на основе принципов нормализации. В рамках данной работы разработан алгоритм приведения схемы БД к 3 НФ (рис. 1 а, б, в, г).

Первые три вложенных цикла (Блоки 3-8) обеспечивают определение всех функциональных зависимостей, существующих между атрибутами отношений БД. Первый, с параметром  $i$  (номер отношения), обеспечивает перебор всех отношений в БД. Вторым и третьим циклы, с параметрами  $j$  и  $z$  (номера атрибутов) соответственно, осуществляют перебор всех пар атрибутов в отдельном отношении, сравнение значений в каждой паре атрибутов, составляют список пар атрибутов (Блок 8), между которыми существуют ФЗ. В список не включаются пары атрибутов с обратными ФЗ.

## II. ВОПРОСЫ ТЕХНИКИ И ТЕХНОЛОГИИ



б



а

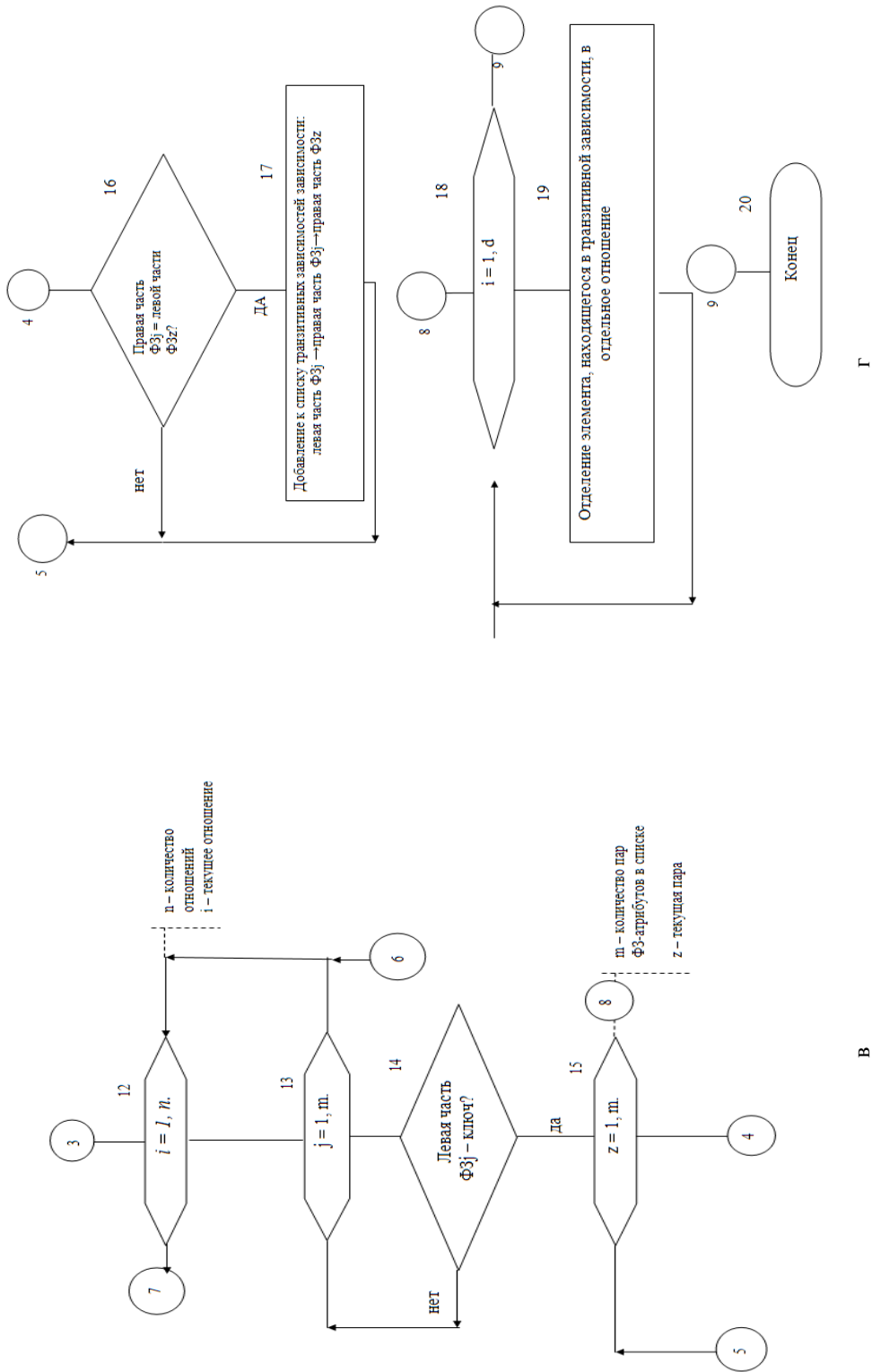


Рис. 1 (а, б, в, г). Алгоритм нормализации

## II. ВОПРОСЫ ТЕХНИКИ И ТЕХНОЛОГИИ

---

Список пар атрибутов должен быть организован как одномерный массив строковых величин. Блок 6 исключает рассмотрение атрибута в паре с самим собой. Блоки 9, 10, 11 исключают из списка пары атрибутов с обратными зависимостями.

Затем начинается участок алгоритма с тремя циклами, вложенными друг в друга (Блоки 12-17). Первый, с параметром  $i$  (номер отношения), обеспечивает перебор отношений, второй, с параметром  $j$  (номер пары ФЗ атрибутов в списке), осуществляет перебор пар атрибутов данного отношения, имеющих ключ в левой части. Третий цикл, с параметром  $z$  (номер пары ФЗ атрибутов в списке), сопоставляет пару, выбранную вторым циклом с остальными парами в списке (Блок 16) и добавляет пары пар ФЗ, удовлетворяющие условиям, в отдельный массив транзитивных зависимостей (Блок 17). Последний цикл (Блоки 18, 19) отделяет второй и третий элемент тройки из исходного отношения в отдельное отношение, удаляя третий элемент из исходного отношения. В Блоке 18 буквой  $d$  обозначено количество транзитивных зависимостей  $i$ -го отношения.

Существующие системы управления базами данных не располагают средствами нормализации, следовательно, проводимые в данной работе исследования являются актуальными. Следующий этап этой работы – реализация разработанного алгоритма на языке высокого уровня с учётом выбора лучшей логической структуры из числа альтернативно возможных. Выбор может быть осуществлён по таким критериям, как требуемый объём памяти и скорость выполнения транзакций.

---

Диго С. М. Базы данных: проектирование и использование. М.: Финансы и статистика, 2005. – 592 с.

Карпова Т. С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.

Краморенко Н. В. Базы данных. Владивосток: Изд-во Дальневост. ун-та, 2004. – 86 с.

URL: <http://ru.wikipedia.org>.