

УДК 004.891: 004.032.26: 004.838.3

UDC 004.891: 004.032.26: 004.838.3

**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ДЛЯ
ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ
ЭФФЕКТИВНОЙ СТРУКТУРЫ**

**INTELLIGENT SYSTEM FOR DATABASE
DESIGN OF EFFECTIVE STRUCTURES**

Малыхина Мария Петровна
к.т.н., профессор

Malykhina Maria Petrovna
Cand.Tech.Sci., professor

Буянов Максим Вадимович
студент
*ФГБОУ ВПО «Кубанский государственный
технологический университет», Краснодар, Россия*

Bujanov Maxim Vadimovich
student
*Kuban State Technological University, Krasnodar,
Russia*

В статье рассматривается автоматизированная система, которая позволит пользователю сделать структуру имеющейся базы данных эффективной путем последовательной нормализации с использованием системы пояснений

The article deals with the automated system that will allow users to make the structure of the existing database efficiently by sequential normalization with the use of explanatory system

Ключевые слова: ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, ЭКСПЕРТНАЯ СИСТЕМА, НОРМАЛЬНЫЕ ФОРМЫ, ЭФФЕКТИВНАЯ СТРУКТУРА, РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Keywords: INTELLIGENT SYSTEM, EXPERT SYSTEM, NORMAL FORMS, EFFECTIVE STRUCTURE, RELATIONAL DATABASE

Введение. На сегодняшний день в информационных системах по ряду важнейших причин доминирующее положение занимают реляционные базы данных (РБД). Но далеко не каждая база данных обладает такими характеристиками, как отсутствие избыточности и логичность выбора структуры. Поэтому важнейшее место в жизненном цикле базы данных занимает ее проектирование. Структурная часть реляционной базы данных — это конечный (ограниченный) набор взаимосвязанных отношений (специальным образом организованных двумерных таблиц). Отношение используются для представления объектов, а также для представления связей между объектами.

Проектирование базы данных – процесс, требующий значительного внимания со стороны проектировщика, так как ему необходимо учесть все нюансы дальнейшей работы с базой данных. Ошибки, допущенные на этом этапе, могут вызвать сложность и даже невозможность сопровождения базы данных.

Из теории баз данных известно [1,3], что база данных должна обладать минимальной избыточностью, и скорость формирования ответов на запросы должна быть высока. Полностью устранить избыточность невозможно, так как она необходима для связи таблиц друг с другом. Но чем меньше избыточность информации, тем меньше вероятность нарушения целостности базы данных и возникновения аномалий обновления в ней. Чем быстрее база данных формирует ответы на запросы пользователя, тем быстрее он получает требуемые данные, а значит, эффективность его работы возрастает.

В процессе разработки структуры базы данных большую помощь проектировщику могла бы оказать интеллектуальная автоматизированная система [2], которая позволила бы ему принять решение, руководствуясь своими знаниями, опытом и интуицией, в случае ситуации с несколькими возможными вариантами проектирования.

Постановка и решение задачи. Для получения наиболее эффективной структуры данных необходимо, прежде всего, разнести важную для будущей системы информацию по различным отношениям. Следует отметить, что вариантов размещения информации может быть много и среди них нужно выбрать один, в наибольшей степени отвечающий указанной цели:

- обеспечение быстрого доступа к данным;
- исключение ненужного повторения данных, которое может явиться причиной ошибок при вводе и нерационального использования дискового пространства компьютера;
- обеспечение целостности данных таким образом, чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Не подлежит сомнению тот факт, что механизмы и здравого смысла, и интуиции проектировщика будут успешнее работать и выдавать корректный результат в том случае, если при всем этом в основе и того и другого лежат знания, касающиеся методов получения схем отношений с требуемыми свойствами. А если необходимых знаний и опыта не хватает, то в результате больших усилий и затрат могут появиться ненадежные, сложные в сопровождении, неэффективные программные продукты.

Исключить или минимизировать подобного рода негативные последствия можно путем использования при построении реляционной схемы БД специального программного продукта, который:

- с одной стороны, позволит проанализировать совокупность первоначальных отношений на наличие негативных моментов в них и привести реляционную схему к виду, лишенному упомянутых негативов;

- с другой стороны, дружественный интерфейс программного продукта и система пояснений происходящих преобразований позволит начинающему разработчику расширить и углубить свои возможности в области проектирования структур БД [4-6].

Объектом исследования при проектировании базы данных в первую очередь являются функциональные зависимости, многозначные зависимости и зависимости соединения, нахождение и анализ которых необходимо произвести для создания эффективной структуры.

Задачей является создание автоматизированной системы (АС), обладающей достаточной гибкостью, для создания баз данных эффективной структуры. АС должна: иметь возможность работать с целой базой данных, рассматривая ее, как совокупность взаимосвязанных таблиц, так и с каждой таблицей по отдельности, анализировать таблицу на соответствие нормальным формам, комментировать процесс анализа,

предлагать с пояснениями возможные варианты преобразования, иметь инструмент построения схемы текущего состояния базы данных, чтобы пользователь мог визуально следить за преобразованиями структуры базы данных, иметь удобный режим диалога с экспертной системой, позволяющий с помощью ответов пользователя и анализа отношений в базе данных строить граф вывода и выводить результаты в легковоспринимаемом виде.

Для определения наличия функциональной зависимости между атрибутами X и Y система просматривает кортежи отношения и если каждая совокупность кортежей с равными X -значениями имеет также равные Y -значения, то между атрибутами X и Y имеется функциональная зависимость.

Обобщенный алгоритм нахождения ФЗ в отношении представлен на рисунке 1.

Для получения эффективной структуры базы данных необходимо спроектировать схему базы данных, которая будет соответствовать нормальной форме Бойса-Кодда или выше.

Первая нормальная форма. Из теории баз данных известно, что отношение находится в 1NF если значения всех его атрибутов атомарны. Так как мы работаем с реляционными базами данных, то будем считать, что значения в нашей базе данных уже являются атомарными и все отношения уже соответствуют первой нормальной форме.

Вторая нормальная форма. Очень часто первичный ключ отношения включает несколько атрибутов, то есть является составным. В этом случае необходимо убедиться, что отношение находится в 1NF, и каждый неключевой атрибут полностью зависит от первичного ключа.

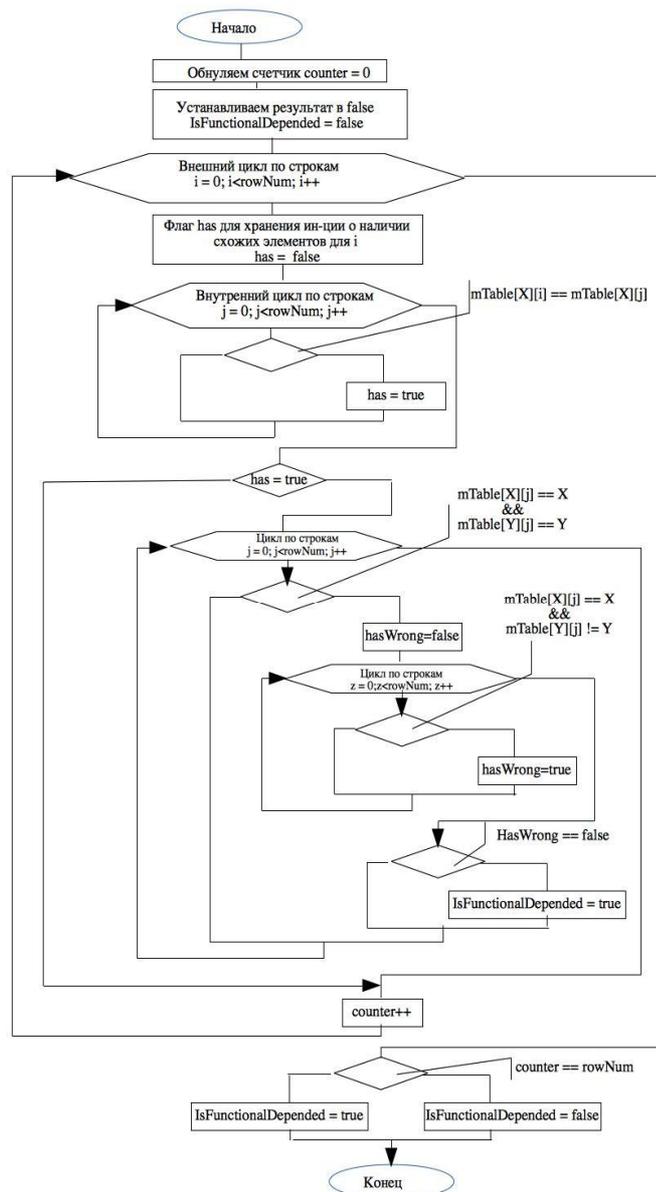


Рисунок 1 - Алгоритм определения наличия ФЗ в отношении

Если в некоторых отношениях обнаружена зависимость атрибутов от части составного ключа, то проводим декомпозицию этих отношений на несколько отношений следующим образом: те атрибуты, которые зависят от части составного ключа, выносятся в отдельное отношение вместе с этой частью ключа.

Рассмотрим отношение **ОРГАНИЗАЦИЯ** (Наименование организации, Город, Код города). Составным ключом является пара

НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ и ГОРОД

Атрибут КОД ГОРОДА зависит от части составного ключа. Имеется функциональная зависимость между атрибутами КОД ГОРОДА и ГОРОД и отсутствует между атрибутами КОД ГОРОДА и НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ. При модификации значений в данном отношении возникнут аномалии обновления и аномалии удаления, поэтому необходимо разбить данное отношение на отношения ОРГАНИЗАЦИЯ (Наименование организации, Город) и ГОРОД (Город, Код города).

Для составного ключа из двух атрибутов можно использовать следующий алгоритм для определения соответствия 2NF (рисунок 2).

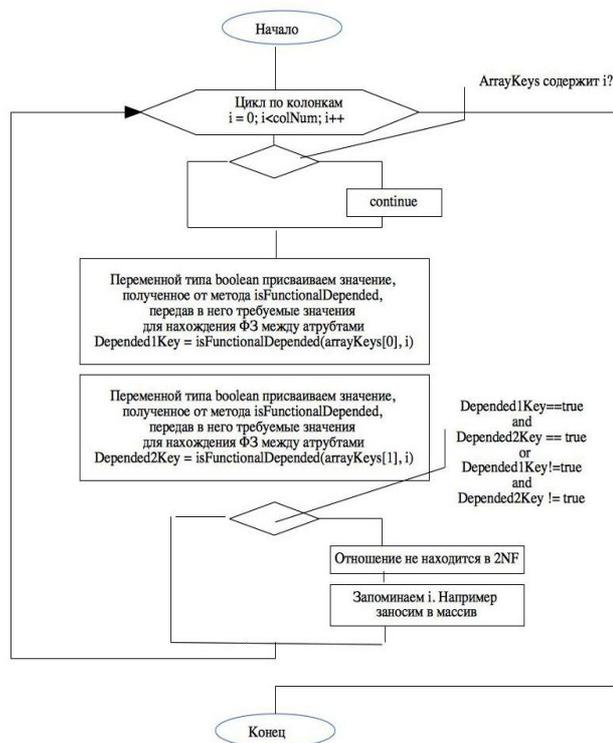


Рисунок 2 - Алгоритм определения соответствия 2NF

Третья нормальная форма. Отношение находится в третьей нормальной форме (3NF) в том и только в том случае, если находится в 2NF и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если в некоторых отношениях обнаружена зависимость некоторых неключевых атрибутов от других неключевых атрибутов, то проводим декомпозицию этих отношений следующим образом: те неключевые атрибуты, которые зависят от других неключевых атрибутов выносятся в отдельное отношение.

Рассмотрим отношение:

ДОГОВОР (Номер договора, Компания, Номер телефона, Тариф, Цена).

В отношении имеются функциональные зависимости между атрибутами КОМПАНИЯ-ТАРИФ и ТАРИФ-ЦЕНА, значит, исходное отношение можно разбить на два отношения ДОГОВОР и КОМПАНИЯ, либо на отношения ДОГОВОР и ТАРИФ. Так как в отношении присутствует несколько функциональных зависимостей, то вариантов декомпозиции будет несколько, поэтому необходимо вмешательство проектировщика, который, на основе своих рассуждений, выберет наиболее предпочтительный, например: ДОГОВОР (Номер договора, Компания, Номер телефона, Цена) и КОМПАНИЯ (Компания, Тариф).

Для определения соответствия 3NF можно использовать следующий алгоритм (рисунок 3).

Нормальная форма Бойса-Кодда. Отношение находится в нормальной форме Бойса-Кодда (BCNF) в том и только в том случае, если каждый детерминант является возможным ключом. Детерминант - это любой атрибут, от которого полностью функционально зависит некоторый другой атрибут.

Ситуация, когда отношение будет находиться в 3NF, но не в BCNF, возникает при условии, что отношение имеет два (или более) возможных ключа, которые, с одной стороны, являются составными, а с другой стороны, перекрываются. Заметим, что на практике такая ситуация встречается достаточно редко, для всех прочих отношений 3NF и BCNF эквивалентны.

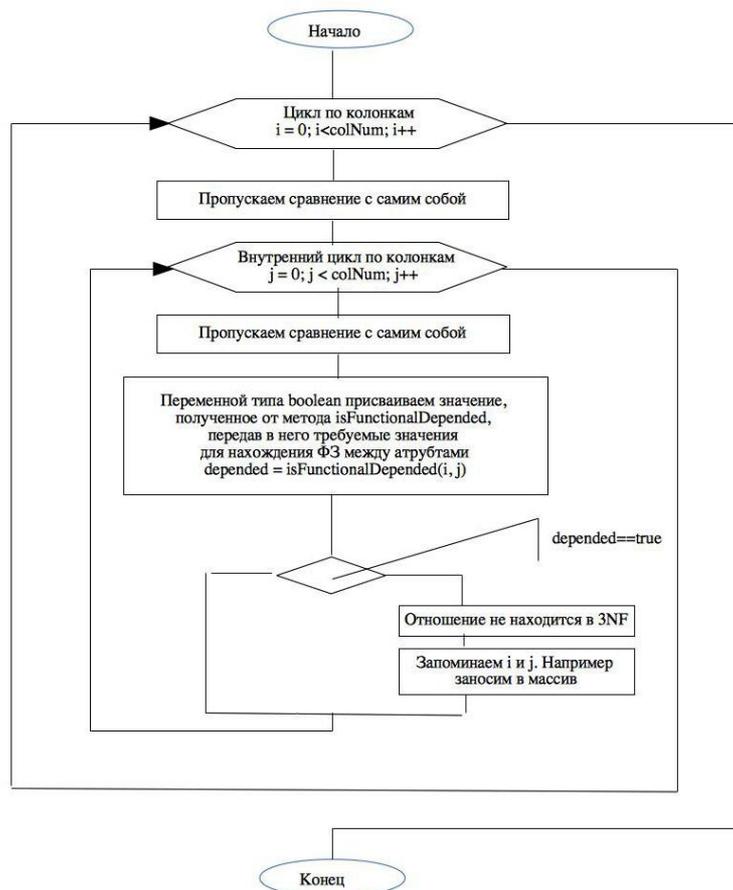


Рисунок 3 - Алгоритм определения соответствия 3NF

Рассмотрим отношение ПЛАТЕЖ (Номер_клиента, ФИО_клиента, Номер_телефона, Сумма, Дата, ФИО_сотрудника). Допустим, что значения атрибута ФИО_клиента уникальны и могут быть использованы наряду с атрибутом Номер_клиента для идентификации клиента. В таком случае можно выделить два составных потенциальных ключа:

(Номер клиента, Номер телефона); (ФИО клиента, Номер телефона).

Декомпозицию требуется провести следующим образом: из исходного отношения убирается и переносится в новое отношение зависимая часть вместе с копией детерминанта. Возникает ситуация, когда требуется вмешательство проектировщика, ведь в качестве внешнего ключа в исходной таблице можно оставить либо НОМЕР КЛИЕНТА либо ФИО КЛИЕНТА:

- 1 вариант: ПЛАТЕЖ (Номер клиента, Номер телефона, Сумма, Дата, ФИО сотрудника) и КЛИЕНТ (Номер клиента, ФИО клиента);

- 2 вариант: ПЛАТЕЖ (ФИО клиента, Номер телефона, Сумма, Дата, ФИО сотрудника) и КЛИЕНТ (Номер клиента, ФИО клиента).

Четвертая нормальная форма. Четвертая нормальная форма касается отношений, в которых имеются повторяющиеся наборы данных. Декомпозиция, основанная на функциональных зависимостях, не приводит к исключению такой избыточности. В этом случае используют декомпозицию, основанную на многозначных зависимостях. Примером отношения, которое не будет соответствовать 4NF, может стать отношение НИР (Номер_НИР, Сотр, Задание_НИР). В такой ситуации единственно возможным ключом отношения является составной атрибут:

(Номер_НИР, Сотр, Задание_НИР).

Полученное в результате отношение характеризуется значительной избыточностью и приводит к возникновению аномалий обновления из-за наличия многозначных зависимостей, которые не являются функциональными.

В отношении существуют следующие две многозначные зависимости:

Номер_НИР -> -> Сотр; Номер_НИР -> -> Задание_НИР.

Поскольку проблема многозначных зависимостей возникает в связи с многозначными атрибутами, то решить ее можно, поместив каждый многозначный атрибут в свою собственную таблицу вместе с ключом, от которого атрибут зависит, то есть разбить его на два отношения СОТРУДНИК(Номер_НИР,Сотр) и ЗАДАНИЕ(Номер_НИР, Задание_НИР).

Пятая нормальная форма. Отношение находится в 5НФ тогда и только тогда, когда любая зависимость по соединению в нем определяется только его возможными ключами. Соответствие 5НФ требует тщательного анализа, и в данной работе не было рассмотрено.

Реализация. Для реализации автоматизированной системы была использована технология JavaFx 2.2 и СУБД MySQL.

JavaFX — платформа для создания Rich Internet application (RIA), позволяет строить унифицированные приложения с насыщенным графическим интерфейсом пользователя для непосредственного запуска из-под

операционных систем, работы в браузерах и на мобильных телефонах, в том числе, работающих с мультимедийным содержимым.

Использование JDBC - платформенно-независимого промышленного стандарта взаимодействия Java-приложений с различными СУБД – предоставляет легкость разработки, возможность перехода на новую базу данных без изменения написанного кода, подключение к любой базе данных с помощью URL. Для модификации существующей базы данных был применен инструмент Liquibase, позволяющий с помощью автоматически создаваемых скриптов изменять структуру базы данных.

Заключение. Данная работа позволяет оценить возможность и выгоду от использования автоматизированной системы, в которой реализованы алгоритмы поиска функциональных зависимостей, определения и анализа таблиц баз данных на соответствие 1NF-5NF. Автоматизированная система значительно может упростить и ускорить работу проектировщика. Более того, был разработан программный продукт, позволяющий применить описанные алгоритмы на практике, в результате чего был получен прототип автоматизированной системы.

Литература

1. Малыгина М.П. Базы данных: основы, проектирование, использовани. Учебное пособие, 3-е изд. – СПб.: БХВ-Петербург, 2007. 528 с.
2. Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. . – СПб. БХВ-Петербург, 2003. 608 с.
3. Томас Коннолли, Каролин Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Вильямс, 2003. 1436 с.
4. Малыгина М.П., Бегман Ю.В. Нейросетевая экспертная система на основе прецедентов для решения проблем обслуживания абонентов сотовой сети. Известия вузов. Северо-кавказский регион. Технические науки. – Новочеркасск. №3. 2009. С. 6 – 9.
5. Оценка эффективности гибридизации интеллектуальных методов на примере нейросетевой экспертной системы на основе прецедентов/ Малыгина М.П., Бегман Ю.В. // Научный журнал КубГАУ [Электронный ресурс]. – Краснодар: КубГАУ, 2013. - № 86(02).– Режим доступа: <http://ej.kubagro.ru/2013/02/pdf/24.pdf>.
6. Симанков В. С., Частикова В. А. Генетические алгоритмы и поиск оптимальных решений // Автоматизация и современные технологии. 2003. №6. С. 39-45.

References

1. Malyhina M.P. Bazy dannyh: osnovy, proektirovanie, ispol'zovani. Uchebnoe posobie, 3-e izd. – SPb.: BHV-Peterburg, 2007. 528 s.
<http://ej.kubagro.ru/2013/05/pdf/41.pdf>

2. Chastikov A.P., Gavrilova T.A., Belov D.L. Razrabotka jekspertnyh sistem. Sreda CLIPS. . – SPb. BHV-Peterburg, 2003. 608 s.
3. Tomas Konnolli, Karolin Begg. Bazy dannyh. Proektirovanie, realizacija i soprovozhdenie. Teorija i praktika. Vil'jams, 2003. 1436 s.
4. Malyhina M.P., Begman Ju.V. Nejrosetevaja jekspertnaja sistema na osnove precedentov dlja reshenija problem obsluzhivaniya abonentov sotovoj seti. Izvestija vuzov. Severo-kavkazskij region. Tehnicheskie nauki. – Novocherkassk. №3. 2009. S. 6 – 9.
5. Ocenka jeffektivnosti gibridizacii intellektual'nyh metodov na primere nejrosetevoj jekspertnoj sistemy na osnove precedentov/ Malyhina M.P., Begman Ju.V. // Nauchnyj zhurnal KubGAU [Jelektronnyj resurs]. – Krasnodar: KubGAU, 2013. - № 86(02).– Rezhim dostupa: <http://ej.kubagro.ru/2013/02/pdf/24.pdf>.
6. Simankov V. S., Chastikova V. A. Geneticheskie algoritmy i poisk optimal'nyh reshenij // Avtomatizacija i sovremennye tehnologii. 2003. №6. S. 39-45.