Проектирование баз данных на основе доменно-ключевой нормальной формы

А.А. Алтайбек
Казахский национальный университет им. аль-Фараби, Алматы, Казахстан
e-mail: Aizhan.Altaibek@kaznu.kz

Аннотация

В статье предлагается методика проектирования логической структуры баз данных (БД). Предлагаемая методика включает разработку концептуальной модели с помощью модели Сущность-Связь и создание логической модели, где отношения приведены к доменно - ключевой нормальной форме посредством выполнения четырех этапов. Описан процесс реализации каждого этапа, где введены два ключевых понятия: отношения-сущности и отношения-связывания. В статье также показаны результаты сравнительного анализа с другими методиками проектирования БД.

Ведение

Структура БД является моделью предметной области, разрабатывающейся в фазах концептуального, логического и физического проектирования данных. Концептуальное и логическое проектирование базы данных - важные этапы успешности разрабатываемой информационной системы.

С помощью предлагаемой методики создается структура БД, удовлетворяющая не только таким критериям как целостность, производительность и отсутствие избыточности, но также удовлетворяющая критерию *расширяемость*. Критерий *расширяемость* является очень важным критерием, который помогает сохранить информационную систему в актуальном и востребованном состоянии на протяжении длительного времени при изменчивости требований предметной области.

Обзор и постановка задачи

Концептуальная модель данных помогает понять общую схему работы определенной предметной области, и строится на основе информации, записанной в спецификациях требований пользователя. Создание концептуального представления БД включает определение важнейших сущностей, его свойств и связей между ними, и выполняется посредством определенного метода моделирования. Модель Сущность-Связь наиболее популярна для разработки концептуальной модели данных. Именно этот метод для создания концептуальной модели будет применяться в нашей работе.

На этапе логического проектирования каждое отношение нормализуется с помощью процесса нормализации. Процесс нормализации основывается на понятиях *функциональной зависимости* и первичного *ключа*. *Функциональная зависимость* - это связь между атрибутами. Если с помощью значения атрибута A, мы сможем вычислить значение атрибута B, тогда атрибут B функционально зависит от атрибута A. Первичный *ключ*- это группа из одного или более атрибутов, которая уникальным образом идентифицирует строку(кортеж) отношения. К сожалению, далеко не все отношения в своем первоначальном виде являются желательными для хранения и обработки данных. А именно, встречаются такие проблемы, как *избыточность данных* и *аномалии модификации*. *Избыточность данных* заключается в наличии в отношениях одинаковой или лишней информации. Минимизация или устранения избыточности данных, прежде всего, сокращает объем памяти БД и повышает производительность

обработки данных. При работе с отношениями, содержащими избыточные данные, могут возникнуть проблемы, которые называются *аномалиями модификации*. Аномалии модификации обнаруживались и устранялись теоретиками реляционных баз данных посредством нормальных форм. С каждым выходом нормальных форм отношения совершенствовались. Хотя нормальные формы помогали, не было теории, гарантирующей, что какая-либо из этих форм устранит все аномалии: каждая форма могла устранить только определенные их виды. Эта ситуация разрешилась в 1981 году, когда была введена доменно-ключевая нормальная форма (ДКНФ)[2]. Отношение находится в ДКНФ, если каждое ограничение целостности отношения является следствием ограничений целостности доменов и ключей [5]. В ДКНФ используются три термина: *ограничение*, *ключ* и *домен*. *Ограничение* - это правило, регулирующее возможные статические значения атрибутов и достаточно точное для того, чтобы можно было установить, выполняется оно или нет. Как уже было определено, *ключ* - уникальный идентификатор кортежа отношения.

Домен - это описание допустимых значений атрибута.

До введения ДКНФ теоретикам реляционных баз данных приходилось продолжать поиск все новых и новых аномалий и нормальных форм. Доказательство Фагина упростила ситуацию: если отношение приведено в ДКНФ, тогда можно быть уверенным, что в нем не будет аномалий модификаций.

Как было отмечено в начале данного раздела, существует проблемы при изменении структуры БД согласно новым требованиям предметной области. Распишем главные проблемы:

- возможность возникновения избыточности данных и аномалий модификаций;
- снижение производительности обработки данных;
- увеличения объема памяти хранения данных;
- трудоемкое изменение запросов, согласно модифицированной структуре БД;
- возможность отказа от текущей структуры, и проектирования новой структуры, что ведет к разработке практически новой информационной системы.

Причина возникновения указанных проблем - очень низкая расширяемость структуры БД. Следовательно, возникает задача в разработке структуры БД, удовлетворяющая не только таким важным критериям, как *целостность* , *производительность*, *отсутствие избыточности и аномалий*, но также не менее важному критерию как *расширяемость*. Главная задача предлагаемой методики - разработать структуру БД, удовлетворяющая всем перечисленным критериям, в том числе и критерию *Расширяемость*.

Определение и теоретико-множественное обозначение объектов реляционной БД

Схемой отношения R называется конечное множество имен атрибутов $\{A_1,A_2,...,A_n\}$. Каждому имени атрибута A_i ставится в соответствие множество D_i называемое *доменом* атрибута A_i, $1 \leq i \leq n$. Домен атрибута A_i, будем также обозначать как $dom(A_i)$. Пусть $\mathbf{D} = D_1 \cup D_2 \cup ... \cup D_n$ *Отношение* r со схемой R - это конечное множество отображений $\{t_1,t_2,...,t_p\}$ из R в $\mathbf{D}$, где каждое отображение $t \in r$ должно удовлетворять следующему ограничению: $t(A_i)$ принадлежит D_i, $1 \leq i \leq n$. Эти отображения называются *кортежами*. A-значение кортежа t - это конкретное значение кортежа t на атрибуте A, и обозначается как $t(A)$ [5].

Ключ отношения r со схемой R является подмножеством $K = \{B_1, B_2, ..., B_m\} \subseteq R$, где для любых двух различных кортежей t_1 и t_2 в r существует такое $B \in K$, что $t_1(B) \neq t_2(B)$[5].

В предлагаемой методике вводятся два ключевых объекта: *отношение - сущность* и *отношение - связывания*. *Отношение-сущность* - это такое отношение, которое не содержит в наборе своих атрибутов атрибуты других сущностей. Другими словами можно сказать, что отношение-сущность не может ссылаться на другие отношения, но другие отношения могут ссылаться на это отношение. *Отношение- связывания* служит для связывания нескольких отношений-сущностей и может содержать дополнительные атрибуты.

В предлагаемой методике множество атрибутов имеют три вида: $P = \{P_1, P_2, ..., P_k\}$ - множество собственных атрибутов, где все атрибуты определяют только одну конкретную сущность; $C = \{C_1, C_2, ..., C_m\}$ - множество связывающих (ссылочных) атрибутов, состоящее только из ключей разных сущностей; $S = \{S_1, S_2, ..., S_l\}$ - множество дополнительных атрибутов, наличие, которых бывает необходимо для удовлетворения определенных требований предметной области. Учитывая данное разделение множеств атрибутов вводим следующие два определения.

Определение 1. *Отношение-сущность r_e со схемой R_e с множеством атрибутов A_e- это конечное множество отображений $\{t_{e_1}, t_{e_2}, ..., t_{e_p}\}$ из R_e в $\mathbf{D_e}$, где каждое отображение $t_e \in r_e$ должно удовлетворять ограничению $t_e(A_{e_i})$ принадлежит D_{e_i} , $1 \le i \le n$, а множество атрибутов A_e удовлетворять ограничению $A_e = P_e$, где P_e - множество собственных атрибутов для отношения r_e.*

Определение 2. *Отношение-связывания r_c со схемой R_c с множеством атрибутов A_c- это конечное множество отображений $\{t_{c_1}, t_{c_2}), ..., t_{c_p}\}$ из R_c в $\mathbf{D_c}$, где каждое отображение $t_c \in r_c$ должно удовлетворять ограничению $t_c(A_{c_i})$ принадлежит D_{c_i} , $1 \le i \le m$, а множество атрибутов A_c удовлетворять ограничению $A_c = C_c$ или $A_c = \{C_c, S_c\}$, где C_c-множество ссылочных атрибутов, S_c-множество дополнительных атрибутов для отношения r_c.*

Методика проектирования БД

Предлагаемая методика построения структуры БД включает концептуальное и логическое проектирования данных. Концептуальная модель БД получается с использованием метода *Сущность-Связь*, где определяются основные сущности и связи между ними. Логическая модель данных получается путем преобразования модели Сущность-Связь в модель, состоящую только из отношений: отношения - сущность и отношения - связывания, а также выполнения ограничений на домены и ключи, согласно ДКНФ. Назовем предлагаемую *модель ОС2*.

Согласно модели Сущность-Связь, объекты, учет которых хотят вести пользователи, представляются *сущностями*, а взаимоотношения между этими сущностями представляются явно определенными *связями*. Данная модель должна определить и предоставить все сущности $e_1, e_2, ..., e_n$ предметной области, с указанием множество атрибутов $A_{e_1} = \{A_{e_{11}}, A_{e_{12}}, ..., A_{e_{1k}}\}$, $A_{e_2} = \{A_{e_{21}}, A_{e_{22}}, ..., A_{e_{2l}}\}$, ..., $A_{e_n} = \{A_{e_{n1}}, A_{e_{n2}}, ..., A_{e_{nm}}\}$ для каждой из этих сущностей, а также, связи $c_1, c_2, ..., c_p$ между указанными сущностями. В данном случае нет необходимости указывать, как именно связаны сущности, достаточно просто определить их наличие между этими сущностями. Когда модель Сущность-Связь готова приступаем к процессу преобразования ее в модель ОС2, состоящий из следующих четырех этапов.

1-ЭТАП: преобразование сущностей в отношения

Пусть $e_1, e_2, ..., e_n$ будут сущностями с множеством атрибутов $A_{e_1} = \{A_{e_{11}}, A_{e_{12}}, ..., A_{e_{1k}}\}$, $A_{e_2} = \{A_{e_{21}}, A_{e_{22}}, ..., A_{e_{2l}}\}$, ..., $A_{e_n} = \{A_{e_{n1}}, A_{e_{n2}}, ..., A_{e_{nm}}\}$ полученные из модели Сущность - Связь. Все выделенные сущности преобразовываем в отношения отношения $r_1, r_2, ..., r_n$ со схемой R, где множество атрибутов $A_1 = \{A_{1_1}, A_{1_2}, ..., A_{1_k}\}$, $A_2 = \{A_{2_1}, A_{2_2}, ..., A_{2_l}\}$, ... , $A_n = \{A_{n_1}, A_{n_2}, ..., A_{n_m}\}$.

2-ЭТАП: декомпозиция отношений на отношения - сущности и отношения - связывания

На этом этапе необходимо исследовать каждое полученное отношение r, связанное с другими отношениями на наличие пересекающихся атрибутов. Главная задача - определить в анализируемом отношении r атрибуты, ссылающиеся на другие отношения. Такие ссылочные атрибуты определяем с помощью оператора пересечения схем или атрибутов отношений. Пусть K будет множеством ключей для отношения $r(A)$ со схемой $R = \{A\} = \{A_1, A_2, ..., A_l\}$, где A - конечное множество атрибутов, включающее множество ключей $K \subseteq A$. Пусть $r_i(A_i)$, со схемой $R_i = \{A_i\} = \{A_{i_1}, A_{i_2}, ..., A_{i_l}\}$ и $K_i \subseteq A_i$ и $r_j(A_j)$, со схемой $R_j = \{A_j\} = \{A_{j_1}, A_{j_2}, ..., A_{j_p}\}$ и $K_j \subseteq A_j$ будут любыми не одинаковыми отношениями, связь которых видна из модели Сущность-Связь, тогда декомпозиция выполняется по следующему правилу: $r_i(A_i), r_j(A_j), i, j > 0, i \neq j : A_{i_k} \cap A_{j_m} \neq \emptyset \Rightarrow A_i - A_{i_k} = E_e$, где $1 \leq k \leq l$, E_i- множество собственных атрибутов отношения r_i. Другими словами, если в наборе атрибутов A_i отношения r_i существует атрибут $\{A_{i_k}\}$, принадлежащий набору атрибутов A_j отношения r_j, то необходимо удалить атрибут $\{A_{i_k}\}$ из A_i.

В результате мы получаем *отношение-сущность* $r_i(A_i)$ со схемой $R_i = \{A_i = E_i\}$, где E_i- множество собственных атрибутов для отношения r_i и другое *отношение-сущности* $r_j(A_j)$, которое изначально содержало множество собственных атрибутов, а связь между отношениями - сущностями r_i и r_j реализуем с помощью введения нового *отношения - связывания* r_c со схемой $R_c = \{C_c\}$ либо со схемой $R_c = \{C_c, S_l\}$, где C_c- множество ссылочных атрибутов, состоящий из набора ключей $C_c = \{K_i, K_j\}$, а S_l- множество дополнительных атрибутов, наличие которого зависит от конкретных требований. Следовательно, отношения r_i и r_j были декомпозированы на две отношения-сущности и отношения-связывания.

Основная наша цель - это добиться максимальной расширяемости структуры БД в будущем. Если придерживаться этой цели, тогда необходимо все отношения-сущности связать с помощью отношений-связывания.

3-ЭТАП: выполнение ограничений на домены и ключи

Ограничением на ключи для всех отношений является их уникальность. Ограничения на домены определяются с помощью их физического и логического (семантического) описания. Физическое описание - это множество значений, которое может принимать атрибут, а логическое описание - это смысл данного атрибута. Ограничения ключей выполняются на уровне СУБД, а ограничения доменов можно выполнить и на уровне СУБД, и на уровне программного кода.

4-ЭТАП: установление связей между всеми отношениями

Данный этап является завершающим этапом предлагаемой методики, где устанавливаются связи между всеми отношениями. На этом этапе необходимо реализовать объединение всех отношений-сущности
посредством указания связей с отношениями- связывания.

В целом метод предлагает преобразование концептуальной модели сущность-связь в модель, представляющая собой взаимосвязь элементарных абстрактных объектов двух типов, описываемых отношением-сущность и отношением-связывания которые удовлетворяют ДКНФ.

Сравнительный анализ и оценка

Хотя предложенная методика может применяться для любой предметной области, для сравнения мы выбрали конкретную область применения для лучшего понимания и предоставления методики. Рассматриваемая предметная область является учебный процесс по кре-

дитной технологии обучения, которая наиболее подвержена к различным изменениям именно в процессах и задачах, что ведет к существенному изменению структуры баз данных (БД). Данная область наиболее подходящая для отображения решаемых проблем, и применения предложенной методики разработки БД.

a) Описание главных требований предметной области

Основная задача учебного процесса по кредитной технологии состоит в развитии у обучающихся способностей к самоорганизации и самообразованию на основе выборности образовательной траектории в рамках регламентации учебного процесса и учета объема знаний в виде кредитов. Так как учебный процесс состоит из многочисленных требований, правил, задач и функций, невозможно описать модель в рамках данной статьи, и для упрощения, мы будем рассматривать только следующие процессы кредитной технологии: *регистрация студентов на дисциплины*, *разработка индивидуального учебного плана (ИУП) студента*, *распределение студентов на группы*. Опишем общие требования к выделенным процессам:

- специальности распределяются на факультеты
- студент обучается только в одной специальности
- для каждой специальности существует общий учебный план (ОУП), который может меняться в зависимости от года поступления
- учебный план состоит из обязательного компонента и компонента по выбору
- студенты регистрируются на дисциплины, указанные в компоненте по выбору
- для каждой зарегистрированной дисциплине формируют группу по выбору
- студент может иметь несколько групп по выбору
- студент распределяется только на одну академическую группу
- ИУП студента формируется на основе выбранных им дисциплин

Согласно указанным требованиям разработана логическая структура БД с применением предлженной методики проектирования.

b) Анализ и результаты

Если для данной предметной области применить традиционную теорию нормальных форм, не включающая ДКНФ, то согласно требованиям все отношения были бы приведены к 4-НФ. Возможно, структура будет оптимальной, но только до введения изменений в эту структуру, которые образовались согласно новым или изменённым требованиям предметной области. Каждая нормальная форма, кроме ДКНФ, ориентируется на требования во время разработки структуры, тем самым структура получается абсолютно не подготовленной к будущему редактированию. Так как ДКНФ справляется с данной проблемой, этот вид нормальной формы является основой предложенного подхода.

Сделан сравнительный анализ традиционной и предлагаемой методик проектирования. Для сравнения мы разработали другую структуру БД, применяя традиционные методы нормализации. Разработка завершилась на 4-НФ. Для того, чтобы показать как работает расширяемость структуры БД, мы устранили из требований *Группу по выбору*, которая создается на основе выбранных студентами дисциплин. Так как из исходных требований мы убрали один тип группы, следовательно, по классическому принципу нормализации, сущность ГРУППА присутствует в сущности СТУДЕНТ. Допустим, прошло некоторое время после внедрения

Таблица 1: Результаты сравнительного анализа

Критерий	Структура, разработанная на основе предложенной методики с применением ДКНФ	Структура, разработанная на основе нормальных форм, кроме ДКНФ
Объем физической памяти БД	16064 KB	21440 KB
Время обработки запроса (был задан запрос на вывод фамилии студента, его академической группы и его всех групп по выбору)	Максимум 1 секунда	1 минута 21 секунды

информационной системы, и в требованиях предметной области появляется новый тип группы - *Группа по выбору*. Если учитывать, что данное изменение не требует переразработки структуры БД, то существует два решения данной проблемы:

1. Добавить атрибут *Группа по выбору* в отношение СТУДЕНТ
2. Либо добавить еще одно отношение, которое свяжет отношение СТУДЕНТ с новым типом группы

Как показало исследование, ни одно из вышесказанных решений не приведет к утешительным результатам, а наоборот, приведет к следующим недостаткам:

1. Увеличение объема памяти хранилища данных
2. Трудоемкое изменение запросов
3. Снижение производительности обработки данных
4. Появление избыточных данных

Структура, разработанная в предыдущем разделе, была бы подготовленной, и данное изменение не привело бы ни каким проблемам. Данная структура является более эффективной, не смотря на множество декомпозированных отношений. В таблице 1 показаны результаты сравнительного анализа по результатам исследования на 23500 студентах в БД, с учетом введенного нового требования в предметную область.

В результате предложенная методика не просто покрывает такие основные критерия, как *целостность*, *устранение избыточности и аномалий модификаций*, *повышение производительности обработки данных*, но также обеспечивает расширяемость структуры БД.

Заключение

Как показывают результаты исследования, предложенная методика разработки структуры БД эффективнее, как на стадии разработки, так и на стадии расширения. Тем самым, логическая модель, полученная с применением описанной методики может прослужить в течение большого периода времени.

Литература

[1] *Коннолли К.Т., Бегг К., Страчан А. Базы Данных. Проектирования, реализация и сопровождение. Теория и практика. - М.: Вильямс, 2000, - 1112 с.*

[2] *Кренке Д. Теория и практика построения Базы данных. - СПб.: Питер, 2003, - 800 с.*

[3] *Codd E.F. A Relational Model of data for Large Shared Databanks //Communications of the ACM, 1970.*

[4] *Codd E.F. Relational Completeness of Data Base Sublanguages // In: R. Rustin (ed.): Database Systems: 65-98, Prentice Hall and IBM Research Report RJ 987, San Jose, California, 1972.*

[5] *Fagin R. A Normal Form for Relational Databases That Is Based On Domains and Keys //CACM Transactions on Databas Systems, 1981.*

[6] *Мейер Д. Теория реляционных баз данных. - М.: Мир, 1987, - 608 с.*