Алеева Ю.Р., Дрождин В.В., Кондрашин М.В. Подсистема реорганизации базы данных гибкой информационной системы поддержки бизнеса **//**  Университетское образование (МКУО-2014): сб. ст. XVIII Междунар. науч.-метод. конф. – Пенза : Изд-во ПГУ, 2014. – С. 151 – 153.

**Подсистема реорганизации базы данных
гибкой информационной системы поддержки бизнеса**

Ю.Р. Алеева, В.В. Дрождин, М.В. Кондрашин

Пензенский государственный университет, Пенза, Россия

Управление современным бизнесом – это сложный процесс, требующий принятия важных управленческих решений в сжатые сроки, основываясь на анализе большого объема разнообразной информации. Для современного руководителя важно уметь не только быстро решать возникающие проблемы, но и своевременно предотвращать их, направляя бизнес в сторону новых возможностей и благоприятных перспектив.

В настоящее время для автоматизированной поддержки бизнеса широко используются системы поддержки принятия решений (СППР), основу которых составляют информационные системы, обеспечивающие сбор, хранение и обработку больших объемов информации. Однако все существующие системы, с одной стороны, недостаточно интеллектуальны, а с другой – очень трудоемки для модификации и включения новых функций. Поэтому именно жесткость и сложность модификации СППР часто ограничивают гибкость бизнеса [1].

Любая СППР содержит подсистему данных, состоящую из двух компонентов: базы данных (БД) и системы управления базой данных (СУБД). Организация больших объемов информации существенно влияет на скорость обработки данных и эффективность всей системы в целом, поэтому БД должна быть эволюционной и способной самостоятельно совершенствоваться в процессе существования. Для разработки механизма автоматической реорганизации БД необходимо выбрать модель данных, разработать алгоритмы выявления функциональных зависимостей (ФЗ) и проектирования на их основе БД в определенной нормальной форме.

Выбор модели данных будем основывать на следующих положениях: БД будет содержать достаточно большой объем информации и должна быть ориентирована на решение широкого круга задач, а в начальный момент времени в системе отсутствует информация о будущем использовании данных. В этих условиях в качестве исходной модели данных целесообразно выбрать реляционную модель данных, являющуюся универсальной моделью данных [2]. В процессе использования системы возможен переход от реляционной модели данных, например, к эволюционной модели данных [3, 4]. Однако выявленные в рамках реляционной модели данных ФЗ будут сохраняться и в других моделях, т.к. они отражают целостность объектов предметной области и отношения между ними, а не отношения между данными в конкретной модели данных. Эффективная обработка данных в БД может осуществляться на основе формальных методов [5].

Рассмотрим выявление функциональных зависимостей. Функциональной зависимостью (ФЗ) множества атрибутов *y* от множества атрибутов *x* называется зависимость *f*: *x* → *y*, устанавливающая, что в любой момент времени для одного значения *x* будет существовать не более одного значения *y*.

Исходными данными для выявления ФЗ является реляционное (универсальное) отношение *R* – таблица из *n* атрибутов и *m* кортежей. Формальный способ выявления ФЗ состоит из следующих шагов.

1) Для каждого атрибута из *R* строим домен значений: из множества различных значений атрибута оставляем только дубликатные значения.

2) Формируем анализируемое отношение *R*' путем замены дубликатного значения номером элемента в домене, уникального значения признаком “уникально”, неопределенного значения признаком “не определено”.

3) Выявление ФЗ с определяющей частью из одного атрибута.

Если атрибут *аi* содержит только уникальные значения (домен *di* = ∅) и нет признаков “не определено”, то существует ФЗ *а*i → *а*1, *а*2, …, *а*i‑1, *а*i+1, …, *аn*, а атрибут *аi* будет являться простым (из одного атрибута) ключом отношения *R*.

Если атрибут *аi* содержит уникальные, дубликатные и неопределенные значения (домен *di* ≠ ∅) и есть атрибуты, разнообразие которых полностью покрывается разнообразием атрибута *аi*, то существует ФЗ *аi* → *а*1, *а*2, …, *аk*.

4) Выявление ФЗ с определяющей частью из нескольких атрибутов осуществляется путем последовательной генерации определяющих частей (сочетаний атрибутов) мощности $l = \overline{2… n-1}$ и определения зависимостей *а*1, *а*2, …, *аℓ* → а*ℓ*+1, *аℓ*+2, …, *аk*.

Анализируются только определяющие части, редуцированные слева (не содержащие транзитивно зависимых атрибутов). Для анализируемой определяющей части формируется редуцированная справа определяемая часть: из атрибутов отношения *R* удаляются атрибуты определяющей части и атрибуты, транзитивно зависящие от атрибутов определяющей и определяемой частей.

Анализ на наличие ФЗ заключается в покрытии разнообразий атрибутов определяемой части разнообразием множества атрибутов определяющей части.

Приведенный способ позволяет выявлять минимальное множество ФЗ, редуцированных слева и частично редуцированных справа (вследствие случайности порядка выявления ФЗ), обладающее следующими свойствами:

- выявленные ФЗ нетривиальны – не выводимы из других зависимостей;

- множество ФЗ является полным – для данного набора нетривиальных ФЗ в универсальном отношении *R* все остальные зависимости являются тривиальными (выводимыми из имеющихся зависимостей).

Для реализации рассмотренного способа выявления ФЗ разработан алгоритм, обладающий следующей сложностью:

- емкостная сложность

*E* = *O*(4⋅*m*⋅*n*) + *O*(16⋅*m*⋅*n*) + *O*(*m*⋅*n*/8 + 12⋅*n*) + *O*(*m*⋅*n*/8 + 16⋅*n*) ≈ *O*(*m*⋅*n*);

- временная сложность

*T* = *O*(2⋅*m*⋅*n*) + *O*(*n*2⋅(*m*/32 + 2⋅*m*)) + *O*((2*n* – n)⋅(*k*⋅*n*/16 + *m*⋅*n*/32 + 3⋅*m*⋅(*n-ℓ*))) ≈ *O*((2*n* – n)⋅(k + *m*)⋅*n*)),

где *k* – количество выявленных ФЗ.

Алгоритм выявления ФЗ имеет низкую емкостную сложность и, в общем случае, экспоненциальную временную сложность, пропорциональную $2^{n}$.

Для понижения временной сложности алгоритма выявления ФЗ целесообразно учитывать следующие факторы:

- в теории баз данных принято ограничивать ключи отношений 6-8 атрибутами, что соответствует идентификации реальных объектов небольшим количеством свойств и снижает временную сложность алгоритма до пропорциональной ($2^{n} – 2^{n-6}$) или ($2^{n} – 2^{n-8}$);

- если в *R* выявлена ФЗ с определяющей частью из *k* атрибутов, то временная сложность алгоритма становится пропорциональной ($2^{n} – 2^{n-k}$);

- если в отношении *R* выявлены две ФЗ с мощностями определяющих частей *k*1 и *k*2 соответственно и *ℓ* общими атрибутами в них, то временная сложность алгоритма становится пропорциональной ($2^{n} – 2^{n-k\_{1}} –2^{n-k\_{2}+l}$).

Следовательно, если в отношении *R* выявляются ФЗ с определяющей частью из 1 или 2 атрибутов, и таких зависимостей достаточно много (соответствует хорошо структурированной предметной области), то временная сложность алгоритма выявления ФЗ становится вполне приемлемой.

Проектирование схемы БД будем осуществлять в третьей нормальной форме (3НФ), т.к. этого достаточно для обеспечения эффективной обработки данных [6]. БД будет находиться в 3НФ, если все ее отношения будут находиться в 3НФ: в отношениях БД неизбыточные ключи, все неключевые атрибуты зависят от ключа и отсутствуют транзитивные зависимости.

Исходными данными для проектирования схемы БД является множество выявленных ФЗ *F* = {*f* | *f*: *x* → *y*, *x* = {*аi*, *а*2, …, *аℓ*}, y = {*аℓ*+1, *аℓ*+2, …, *аk*}, *X* = {*x*, *y*}, *X* ⊆ *R*}. Формальный способ проектирования БД состоит из следующих шагов.

1) Для каждой ФЗ *f*: *х* → *у* вычисляется замыкание атрибутов *Х*+ = *Х* ∪ *z*, если существует зависимость *f*': *х*' → *z* такая, что *х*' ⊆ {*х*, *у*}. Вычисление замыкания атрибутов для зависимости *f* осуществляется до тех пор, пока есть *f*' ∈ *F*. Построенное множество расширений ФЗF+ = {*f* | *f*: *х* → *у*, *X* = {*x*, *y*}, *Х* = *Х*+} будет использоваться для дальнейшего анализа и проектирования БД.

2) Для каждой ФЗ *f* ∈ F+ определяется количество атрибутов *N* = |*Х*|.

3) В F+ выбирается *f* минимальной мощности (*N* = min) и создается отношение *Ri* = (*xi*, *yi*) с ключом *ki* = *xi*. Если в F+ существует зависимость *f*': *х*' → *у*' с *X*' = {*x*', *y*'} такая, что *X*' = *X*, то она проецируется на отношение *Ri*,а идентификатор *x*' становится возможным ключом *ki*' отношения *Ri*. Удаляем из F+ зависимость *f* и зависимости *f*'.

4) Редуцируем зависимости в F+ справа: в зависимостях *f*, содержащих ключ *ki* ∈ {*х*, *у*}, удаляем атрибуты *yi* из *у* (*у* = *у* \ *yi*).

Шаги 2-4 выполняются пока F+ ≠ ∅.

Таким образом, будет сформирована квазиоптимальная схема базы данных в 3НФ.

Алгоритм, реализующий способ проектирования БД в 3НФ имеет следующую сложность:

- емкостная сложность *E* = *O*(*k*⋅*n*/16 + 4⋅*k*) + *O*(*r*⋅*n*/32) ≈ *O*((*k* + *r*)⋅*n*);

- временная сложность *T* = *O*(2⋅*k*2⋅*n*) + *O*(*k*2⋅*n*/32) + *O*(*k*2⋅*n*/16) ≈ *O*(*k*2⋅*n*),

где *k* – количество ФЗ, *r* – количество отношений в БД с новой структурой.

Алгоритм выявления ФЗ имеет невысокую временную и емкостную сложности.

Перенос исходной БД в БД с новой структурой может осуществляться средствами СУБД с помощью SQL-запросов.

Апробация алгоритмов проведена на основе данных из БД информационной системы университета и получены следующие результаты:

- для отношения из 17 атрибутов и 5866 кортежей выявлено 17 ФЗ и сформировано 10 отношений, время проектирования БД 0,7 с.;

- для отношения из 11 атрибутов и 32897 кортежей выявлено 7 ФЗ и сформировано 5 отношений, время проектирования БД 1,2 с.

Предложенный механизм реорганизации БД позволяет формировать и поддерживать корректную и эффективную БД адекватную изменяющейся внешней среде (решаемым задачам).

Литература

1. Тейлор, Д. Почти интеллектуальные системы / Д. Тейлор, Н. Рэйден. – СПб. : Символ-Плюс, 2009. – 448 с.

2. Мейер, Д. Теория реляционных баз данных / Д. Мейер. – М. : Мир, 1987. – 608 с.

3. Дрождин, В. В. Системный подход к построению модели данных эволюционных баз данных / В. В. Дрождин // Программные продукты и системы. –
2007. – № 3. – С. 52–55.

4. Дрождин, В. В. Открытость структур в эволюционной модели данных / В. В Дрождин // Программные продукты и системы. – 2009. – № 2. – С. 135–137.

5. Волчихин, В. И. Логико-алгебраические модели и методы в проектировании функциональной архитектуры распределенных систем хранения и обработки данных / В. И. Волчихин, С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2012. – № 2 (22). – С. 3–16.

6. Дейт, К. Введение в системы баз данных /К. Дейт. – М. : Издательский дом “Вильямс”, 2005. – 1328 с.