Теоретическая модель неподдерживаемой реляционной базы данных

И. А. Тарханов

В статье рассматривается теоретическая модель реляционной базы данных, которая в процессе эксплуатации становится неподдерживаемой. В ведении раскрывается суть понятия неподдерживаемой базы данных (далее БД). Затем рассматриваются как проблемы, возникающие в таких БД, влияют на схему отношений. Далее предлагается использовать вероятностный подход для оценки результатов поиска. В заключении приводится алгоритм поиска, основанный на этой модели.

Введение

Говоря о неподдерживаемой БД, следует раскрыть понятие неподдерживаемой информационной системы (далее ИС). Неподдерживаемая ИС это устаревшая программа, которую по каким-то причинам невозможно модифицировать (использование устаревшего языка программирования и/или СУБД, отсутствие разработчиков и службы технической поддержки, отсутствие исходных кодов ИС и т. д.). База данных такой системы является неподдерживаемой [1]. Перечислим основные трудности, характеризующие неподдерживаемые базы данных:

1. **Частичное отсутствие информации о формате БД**. Отсутствие людей, проектировавших БД, или отсутствие четкого описания ее формата.
2. **Технические и материальные трудности** в процессе эксплуатации БД. В процессе использования БД, можно столкнуться с ограничением максимально возможного размера БД, невозможностью одновременной работы нескольких пользователей или отсутствием средств для дальнейшего развития БД.

3. **Неоднородность наполнения БД**. Отсутствие контроля за единообразным наполнением БД приводит к тому, что БД используется в «разных целях» разными ИС.

Практически для любой информационной системы, работающей с БД, изначально разработанной для другой ИС, эта БД является неподдерживаемой. Ситуация усложняется, если одна и та же схема базы данных используется на разных серверах, например, в региональных представительствах территориально-распределенной компании [1].

Далее будут рассматриваться только реляционные БД, как наиболее распространенные на сегодняшний момент.

1. Неподдерживаемая реляционная модель данных

1.1. Реляционная модель данных

Обратимся к реляционной модели данных. В манипуляционной составляющей реляционной модели данных определяются два базовых механизма манипулирования реляционными данными — основанная на теории множеств реляционная алгебра и базирующееся на математической логике (точнее, на исчислении предикатов первого порядка) реляционное исчисление. В качестве основы в статье будут взяты определения из реляционной алгебры A Дейта и Дарвена, как наиболее приближенной к существующим реализациям SQL.

Пусть r — отношение, A — имя атрибута отношения r, T — имя соответствующего типа (т. е. типа или домена атрибута A), v — значение типа T. Тогда:

- заголовком Hr отношения r называется множество атрибутов, т. е. упорядоченных пар вида $< A, T >$. По определению никакие два атрибута в этом множестве не могут содержать одно и то же имя атрибута A. Далее под схемой отношения будет так же пониматься заголовок отношения;
- кортеж tr, соответствующий заголовку Hr, — это множество упорядоченных триплетов вида $< A, T, v >$, по одному такому триплету для каждого атрибута в Hr;
- тело Br отношения r — это множество кортежей tr. Заметим, что (в общем случае) могут существовать такие кортежи tr, которые соответствуют Hr, но не входят в Br;
- значением Vr отношения r называется пара множеств $< Hr, Br >$;
- переменной Var отношения r называется именованный контейнер, который может содержать любое допустимое значение Vr;

- реляционная база данных D это множество пар $< Var, Hr >$ [2–4].

В качестве базовых операций используем операции алгебры А, которые имеют названия обычных логических операций, чтобы избежать путаницы, имена реляционных операций берутся в угловые скобки: <NOT>, <AND>, <OR> и т. д. В исходный базовый набор операций входят операции реляционного дополнения <NOT>, удаления атрибута <REMOVE>, переименования атрибута <RENAME>, реляционной конъюнкции <AND> и реляционной дизъюнкции <OR>. В третьем манифесте Дейта и Дарвена доказано, что операция <RENAME> и операция <AND> или <OR> могут быть выражены через другие операции реляционной алгебры [2, 4].

1.2. Проблема неадекватного описания предметной области

Описанные выше механизмы реляционной алгебры обладают одним важным свойством: они замкнуты относительно понятия отношения. Это означает, что выражения реляционной алгебры и формулы реляционного исчисления определяются над отношениями реляционных БД и результатом их «вычисления» также являются отношения. В результате любое выражение или формула могут интерпретироваться как отношения, что позволяет использовать их в других выражениях или формулах. Ряд проблем приводят к тому, что хранящаяся в БД информация неадекватно описывает предметную область. Рассмотрим влияние проблем неподдерживаемых БД на схемы (заголовок) отношений более детально.

Частичное отсутствие информации о схеме БД. Необходимо сказать, что под словами «отсутствие информации о схеме БД» подразумевается отсутствие информации у пользователя (информационной системы) о схеме отношений в БД. При этом в самой БД будут корректные и вполне содержательные данные, удовлетворяющие требованиям ее целостности. Покажем это на простом примере: есть отношение СОТРУДНИКИ, схема которого состоит из следующих атрибутов

{НОМЕР_СОТРУДНИКА, ИМЯ_СОТРУДНИКА,
ФАМИЛИЯ_СОТРУДНИКА, ВРЕМЯ_НА_РАБОТЕ}.

В процессе использования БД филиалом А компании возникла потребность в хранении информации о зарплате сотрудников, для этого было добавлен новый атрибут ЗАРПЛАТА. При этом другим отделам, работающим с этой БД ничего про новый атрибут в схеме СОТРУДНИКИ неизвестно. Они могут использовать для тех же самых целей отношение ЗАРПЛАТА{НОМЕР_СОТРУДНИКА, ЗАРПЛАТА}, про которое, в свою очередь ничего не знают в филиале А. Тем самым схема базы данных

становится **избыточна** (появляется дублирующая информация). Проще говоря, в схему БД добавляются новые атрибуты или целые отношения.

Технические трудности. Рассмотрим отношение из предыдущего примера. В филиале Б компании в отношении СОТРУДНИКИ интенсивно обновляется значение атрибута ВРЕМЯ_НА_РАБОТЕ в процессе работы, что затрудняет доступ сразу нескольких пользователей к данным отношения. Для того, чтобы можно было осуществлять работу с именами сотрудников, в БД создается новое отношение ИМЕНА_СОТРУДНИКОВ{ НОМЕР_СОТРУДНИКА, ИМЯ_СОТРУДНИКА, ФАМИЛИЯ_СОТРУДНИКА}, а аналогичные атрибуты в отношении СОТРУДНИКИ либо перестают заполняться и использоваться, либо вообще удаляются из отношения. Очевидны проблемы при работе с этой БД пользователей из других филиалов компании и других ИС, которые ничего не знают про **отсутствие** атрибутов или целых отношений изначально в схеме присутствовавших.

Неоднородность наполнения. Вернемся опять к отношению СОТРУДНИКИ. В филиале В было принято решение, что удобнее использовать для хранения имени, фамилии и отчества сотрудника один атрибут ИМЯ_СОТРУДНИКА. Поэтому домен (тип) атрибута ИМЯ_СОТРУДНИКА меняется. Теперь это не множество всех возможных имен, а множество всех возможных сочетаний фамилии, отчества и имени. Изменение домена, на котором определен атрибут схемы в отношении назовем **искажение** атрибута.

Здесь приведены простые примеры того, что происходит со схемой в неподдерживаемой БД. Любая из перечисленных во введении проблем может привести к добавлению, отсутствию атрибутов и отношений в схеме БД или их искажению (см. рис. 1).

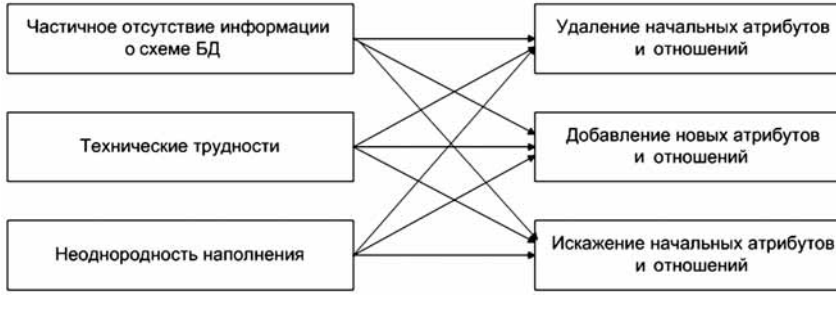

Рис. 1. Связь между проблемами и изменениями схемы в неподдерживаемых БД

1.3. Схема результирующего отношения

Выше были определены «элементарные» действия, которые выполняются со схемой и приводят базу данных в неподдерживаемое состояние. Описывая схему неподдерживаемых баз данных, необходимо знать, как менялась схема всех отношений в базе данных: какие атрибуты обновлялись, удалялись и искажались. Имея «наглядный» формат записи изменений, удобней сравнивать и выявлять все проблемы, возникающие во время эксплуатации БД.

Определение 1. Удаленным атрибутом a^- в заголовке Hr отношения r называется атрибут, который был удален из начального заголовка Hr.

Определение 2. Добавленным атрибутом a^+ в заголовке Hr отношения r называется атрибут, который был добавлен в начальный заголовок Hr отношения.

Перед тем, как определить искажения атрибута, нужно помнить, что атрибут в схеме отношения это пара — имя и домен. Следовательно, изменять можно как домен, так и имя атрибута.

Определение 3. Искаженным атрибутом a^{C} в заголовке Hr отношения r называется атрибут, у которого был изменен тип:

$$Hr< a, T_A > \Rightarrow Hr< a^{\mathrm{C}}, T_A^{\mathrm{C}} >,$$

где $T_A \neq T_A^{\mathrm{C}}$ и $a = a^{\mathrm{C}}$; или атрибут a^R, у которого было изменено имя:

$$Hr\{< a, T_A >\} \Rightarrow Hr\{< a^R, T_A^R >\},$$

где $T_A = T_A^R$ и $a \neq a^R$.

Последовательное изменение заголовка отношения записывается так:

$$Hr\{a, b\} \Rightarrow Hr\{a^R, b\} \Rightarrow Hr\{a^R, b^-\} \Rightarrow Hr\{a^R, b^-, c^+\}$$

— обозначает, что в заголовке Hr сначала переименовали атрибут a, затем удалили атрибут b, а затем добавили атрибут c.

Более подробная запись заголовков вместе с типом:

$$Hr\{< a,T_A >, < b,T_B >\} \Rightarrow Hr\{< a^R,T_A^R >, < b,T_B >\} \Rightarrow$$

$$\Rightarrow Hr\{< a^R,T_A^R >, < b^-,T_B >\} \Rightarrow Hr\{< a^R,T_A^R >, < b^-,T_B >, < c^+,T_C >\},$$

где $T_A \neq T_A^R$.

Удаленным, добавленным или искаженным может быть как один атрибут, так и все атрибуты, входящие в заголовок отношения. Если

с одним и тем же атрибутом происходило последовательно несколько операций, то запись имеет следующий вид:

$$Hr\{a, b\} \Rightarrow Hr\{a^R, b\} \Rightarrow Hr\{a^{(R,C)}, b\}.$$

После удаления атрибута никакая другая последовательная операция с этим атрибутом смысла не имеет. Отметим, что

$$Hr\{a\} \Rightarrow Hr\{a, b^+\} \Rightarrow Hr\{a, b^{(+,-)}\} = Hr\{a\},$$

но при этом

$$Hr\{a, b^-\} \Rightarrow Hr\{a, b^{(-,+)}\} \neq Hr\{a, b\},$$

так как в отношении, соответствующему заголовку справа от $\neq$, кортежи $< b, T_B, v >$ не удаляются в отличие от отношения заголовка слева.

В примерах выше атрибуты в заголовке одного отношения менялись **последовательно**. Но возможна и более сложная ситуация, когда начальную схему используют в физически разных БД (например, в разных филиалах). В этом случае, чтобы отобразить все изменения начальной схемы нужно последовательно учесть изменения во всех заголовках отношений во всех использующих данную схему базах данных.

$$Hr\{a, b, c\}$$

$$\Downarrow$$

$$Hr_1\{a, b^R, c\}\ \ Hr_2\{a, b, c^-\}\ \ Hr_3\{a, b, c, d^+\}$$

$$\Downarrow$$

$$Hr_R\{a, b^R, c^-, d^+\}$$

Начальная схема Hr отношения r использовалась в трех базах данных, где в процессе ее изменения **параллельно** образовалось три новых схемы Hr_1, Hr_2, Hr_3. Затем все изменения всех трех схем были отражены в результирующей схеме Hr_R.

Определение 4. Результирующим заголовком Hr_R отношения r называется заголовок, в котором последовательно отражены все изменения атрибутов (добавление, удаление и искажение) относительно начального заголовка Hr.

Если в процессе эксплуатации БД заголовок отношения не менялся, то $Hr_R = Hr$.

В процессе эксплуатации разных баз данных с одной и той же схемой могут произойти разные изменения одного и того же атрибута

в схеме. В этом случае в результирующий заголовок попадут все изменения. Например,

$$Hr\{a, b\}$$

$$\Downarrow$$

$$Hr_1\{a, b^C\}\ Hr_2\{a, b^-\}$$

$$\Downarrow$$

$$Hr_R\{a, b^{C,-}\}$$

Перед тем, как перейти к непосредственному применению результирующих отношений покажем, что любая операция, отражающаяся в изменении состояний атрибутов в заголовке результирующего отношения справедлива с точки зрения реляционной алгебры и после добавления, удаления и искажения атрибутов отношение остается реляционным.

Теорема 1. *Все операции с атрибутами, отражающиеся в результирующем заголовке отношения, выражаются через базисные операции реляционной алгебры А Дейта и Дарвена.*

Доказательство. Покажем, как через <REMOVE>, <AND> и <NOT> выполнить операции со схемой: удаление, добавление и искажение атрибута (переименования и изменение типа).

Удаление атрибута. Пусть $r < REMOVE > a = s$, где $Hr\{a, b, c\}$, тогда

$$Hs = Hr \text{ minus}\{< a, T_A >\} = Hr\{a^-, b, c\}.$$

Добавление атрибута. Пусть $r_1 < AND > r_2 = s$, где $Hr_1\{a, b\}$ и $Hr_2\{b, c\}$ и любому кортежу $< b, T_B, v >$ из r_1 равен один кортеж $< b, T_B, v >$ из r_2, тогда

$$Hs = Hr_1 \cup Hr_2 = Hr_1\{a, b, c^+\}.$$

При этом, согласно определению $< AND >$, Bs это естественное соединение двух отношений r_1 и r_2.

Переименования атрибута. Пусть

$(r_1 < AND > r_2) < REMOVE > b = s$, где $Hr_1\{a, b\}$ и $Hr_2\{b, c\}$ и любому кортежу $< b, T_B, v >$ из r_1 равен один кортеж $< b, T_B, v >$ из r_2 и в r_2 каждому кортежу $< c, T_C, v_C >$ соответствует кортеж $< b, T_B, v_B >$, у которого $T_B = T_C$, $v_B = v_C$ и $a \neq c$. Тогда

$$Hs = Hr_1\{a, c\} = Hr_1\{a, b^R\}.$$

Bs состоит из всех кортежей $< a, T_A, v_A >$ и соответствующим им $< c, T_B, v_B >$.

Изменение типа атрибута. Пусть $(r_1 < REMOVE > a) < AND > r_2 = s$, где $Hr_1\{a, b, c\}$ и $Hr_2\{a, b\}$ и любому кортежу $< b, T_B, v >$ из r_1 равен один кортеж $< b, T_B, v >$ из r_2 и любому кортежу $< a, T_{A1}, v_{A1} >$ из r_1 равен один кортеж $< a, T_{A2}, v_{A2} >$ из r_2, у которого $T_{A1} \neq T_{A2}$ и $v_{A1} = v_{A2}$, тогда

$$Hs = Hr_1\{a^C, b, c\}.$$

Bs состоит из всех кортежей $< a, T_{A2}, v_{A2} >$ и соответствующим им $< b, T_B, v >$ и $< c, T_C, v_C >$.

Теорема доказана. □

Определение 5. Неподдерживаемой базой данных D называется база данных, в которой в множестве пар $< Var, Hr >$ хотя бы с одним заголовком Hr проводилась операция удаления, добавления или искажения атрибута и для данного заголовка можно построить результирующий заголовок Hr_R, который не равен исходному Hr.

Следствие 1. *Все операции классической реляционной алгебры справедливы для неподдерживаемых отношений, согласно тому, что все операции классической реляционной алгебры Кодда выражаются через базисные операции Агебры А Дейта Дарвена.*

Данное следствие доказано в «Третьем Манифесте» [2, 4].

Теоремой 1 доказано, что любое неподдерживаемое отношение это некоторое обыкновенное отношение, в котором фиксируются все изменения заголовков отношений.

Следствие 2. *Если r отношение с заголовком Hr, а r_R его результирующее отношение с заголовком Hr_R, то $r_R \backslash r = s_1$ и $r \backslash r_R = s_2$, где s_1 и s_2 так же реляционные отношения.*

Возможны случаи, когда при изменении атрибута его конечное или начальное состояние не известно.

Определение 6. Неопределенным атрибутом a^* в заголовке Hr отношения r называется атрибут, про который известен только сам факт его изменения.

Если известен только сам факт изменения атрибута по сравнению с начальной схемой, то его можно отразить в результирующей схеме:

$$Hr\{a, b\} \Rightarrow Hr_R\{a, b^*\}.$$

Если не известно начальное состояние атрибута в схеме, то

$$Hr\{a^*, b\} \Rightarrow Hr_R\{a, b\}.$$

Неопределенность атрибута это «метка», говорящая о том, что частично или полностью отсутствует информация о данном атрибуте. Состояние «неопределенности» атрибута похоже на искажение, но в отличие от искажения может присутствовать и в начальной схеме.

О кортежах, соответствующих результирующему заголовку отношения нельзя делать никаких выводов, так как этот заголовок отражает лишь «историю» изменения атрибутов в нем и никак не описывает значения кортежей для этих атрибутов до и после изменения схемы.

Из рассмотренной в данном разделе теоретической модели неподдерживаемой реляционной базы данных можно сделать следующие выводы:

Частичное отсутствие информации о схеме БД, технические трудности при ее использовании и неоднородность наполнения приводят к изменению схемы. Изменение состоит из нескольких элементарных операций: удаление атрибута, добавление атрибута, искажение атрибута, в свою очередь состоящее из переименования и изменения типа.

Зная начальную схему отношения в неподдерживаемой БД и историю изменения этой схемы, как последовательно (одной и той же БД), так и параллельно (одновременно нескольких) можно построить результирующую схему отношения.

Отсутствие информации о некоторых атрибутах в начальной или в результирующей схеме отношения может быть отражено в схеме с помощью неопределенного атрибута.

Все операции над результирующими схемами отношений могут быть выражены любым языком, полным относительно реляционной алгебры, например, SQL.

Перейдем к применению результирующих схем отношений для поиска в неподдерживаемых БД.

2. Поиск в неподдерживаемой БД

2.1. Вероятность изменения схемы

Определим вероятностное пространство событий изменения результирующей схемы в реляционном отношении (Ω, F, P), где Ω — пространство элементарных исходов, F — множество событий, P — вероятностная мера на F.

Множеством элементарными исходов будут все возможные варианты результирующей схемы отношений. Для $Hr_R\{a, b, \mathrm{c}^R, d^-\}$ это:

$$\{a, b, \mathrm{c}, d\},$$
$$\{a, b, c, d^-\},$$

$$\{a, b, \mathrm{c}^R, d\},$$
$$\{a, b, \mathrm{c}^R, d^-\}.$$

Пусть событие A_i означает, что i-й реквизит изменился и перешел в конкретное состояние. Иными словами, $a_i \Rightarrow a_i^+$ или $a_i \Rightarrow a_i^-$ или $a_i \Rightarrow a_i^R$ или $a_i \Rightarrow a_i^C$ или $a_i \Rightarrow a_i^*$, при $1 \leqslant i \leqslant n$, где n это количество реквизитов в отношении, изменивших свое состояние.

$$P(A_i) = \frac{2^{n-1}}{2^n} = \frac{1}{2}. \quad (1)$$

При этом, если атрибут менялся последовательно и в результирующем отношении имеет следующий вид $a^{(R,\mathrm{C})}$, то вероятность будет вычисляться также

$$P\left(A_i^{(R,\mathrm{C})}\right) = \frac{1}{2},$$

так как события A_i^R и $A_i^{(R,\mathrm{C})}$ происходили последовательно и с точки зрения конечной результирующей схемы это одно событие, приводящие к одному элементарному исходу. В таком случае, предполагается, что после A_i^R всегда происходит $A_i^{(R,\mathrm{C})}$. Если это не так, то у A_i возможны N последовательных состояний. В результирующей схеме может быть последним любое из $A_i^{(1\ldots N)}$ событий. Тогда вероятность событие в одном из последовательных состояний N:

$$P\left(A_i^{(1\ldots N)}\right) = \frac{1}{2^N}. \quad (2)$$

Если атрибут менялся параллельно, то множество элементарных исходов выглядит иначе. Для $Hr_R\{a, b, c^{R,C}, d^-\}$ это:

$$\{a, b, \mathrm{c}, d\},$$
$$\{a, b, c, d^-\},$$
$$\{a, b, \mathrm{c}^R, d\}$$
$$\{a, b, \mathrm{c}^R, d^-\},$$
$$\{a, b, \mathrm{c}^C, d\},$$
$$\{a, b, \mathrm{c}^C, d^-\}.$$

Следовательно,

$$P\left(A_i^{R,\mathrm{C}}\right) = \frac{2}{6} = \frac{1}{3}.$$

В общем виде:

$$P(A_i^j) = \frac{1}{m+1}, \quad (3)$$

где $1 \leqslant j \leqslant m$, m — количество параллельных изменений a_i (всех возможных конечных состояний a_i, кроме начального) в результирующем отношении, а A_i^j событие изменения атрибута i при некотором параллельном использовании j. Для любого Hr_R $A_1, A_2, ...A_n$ события независимые.

Аналогично событиями A_i для искажения, добавления и удаления атрибута можно определить события и для неопределенных атрибутов. Если A_i это событие, при котором атрибут $a_i \Rightarrow a_i^*$, то вероятность события A_i вычисляется по формулам (1), (2), (3). Вероятность события, при котором атрибут в начальной схеме имел состояния неопределенного, а в результирующей стал определенным, вычисляется точно так же.

2.2. Задача поиска

Рассмотрим непосредственно саму задачу поиска в реляционной модели данных в пределах одного отношения r с заголовком Hr.

Объектом поиска является множество атрибутов X, состоящее из одного или нескольких атрибутов и $X \subset Hr$. Результатом поиска является отношение s, заголовок которого состоит из множества атрибутов X. В это отношение попадают все кортежи из r с заголовком X, удовлетворяющие условиям на атрибуты $y_1, y_2, ...y_n$ из множества атрибутов Y, $Y \subset Hr$, $1 \leqslant i \leqslant n$, где n количество атрибутов в Y.

Определим событие S для каждого кортежа tr из r с заголовком Hr. S происходит, если значения tr для атрибутов $y_1, y_2, ...y_n$ из Y удовлетворяют условиям по каждому из этих атрибутов (на самом деле учитываются еще логические операторы между условиями). Событию S поставим в соответствие случайную величину η, $\eta = 1$, если событие выполняется и 0, если нет. Если значения tr для атрибутов из Y не удовлетворяют условиям, тогда событие S не происходит, а происходит несовместное с ним событие $\overline{S}$. В свою очередь, каждому из атрибутов y_i соответствует случайная величина $\eta_i = 1$, если значение из триплета $< y_i, T_i, v_i >$ удовлетворяет условию на y_i, либо 0, если нет. Каждой η_i соответствует гипотеза S_i. Полагаем, что $S_1, S_2, ...S_n$ независимы. Это упрощающее допущение. На самом деле, атрибуты в Y могут иметь функциональную зависимость между собой, но здесь воспользуемся свойствами наивной байесовской классификации для переменных $\eta_1, \eta_2, ...\eta_n$:

- все переменные являются одинаково важными;
- все переменные являются статистически независимыми, т. е. значение одной переменной ничего не говорит о значении другой.

Построим байесовскую сеть для случайной переменной η.

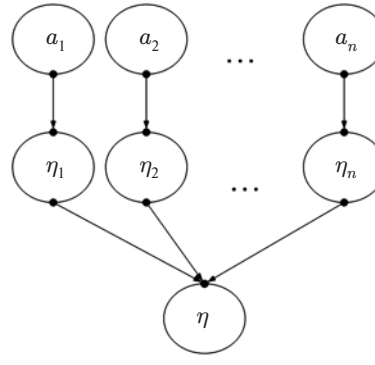

Рис. 2. Байесовская сеть случайных переменных в задаче поиска

При оценке результатов выборки из неподдерживаемой БД необходимо учитывать, что каждая случайная переменная η_i относится к определенному известному состоянию атрибута y_i. В результирующем отношении неподдерживаемой БД состояние атрибута тоже неопределенная величина — a_i. Событие A_i для каждого изменяющегося атрибута результирующего отношения как раз соответствует этой случайной величине. Построим байесовскую сеть для случайной переменной η с учетом, того, что поиск ведется по результирующему отношению.

Далее рассмотрим оценку результатов поиска каждого кортежа в r, вычислив вероятности для всех описанных событий.

2.3. Индекс соответствия запросу

В теории нечетких множеств получило развитие направление нечетких запросов к реляционным базам (fuzzy queries, flexible queries) и нечетких БД (fuzzy DBs), механизмы которых были впервые предложены в 1984 году и впоследствии получили развитие в работах Д. Дюбуа и Г. Прада [6, 7]. Но сами по себе нечеткие БД характеризуются неполнотой, т. е. неточностью, неопределенностью и нечеткостью информации, содержащейся в ней. В отличие от неподдерживаемых БД, формат их известен и нет проблемы неадекватного описания предметной области.

Для оценки результатов нечетких запросов Дюбуа и Г. Прада предложили использовать индекс соответствия запросу (Query Compatibility Index, QCI) вычисляемый для каждой строки в анализируемом отношении. Но технология калькуляции индекса для нечетких данных не подходит для использования в логике запросов к данным. Это логично, потому что теория нечетких множеств вообще не является методом формирования

Таблица 1

Логика вычисления QCI по вероятностям гипотез

Логическая операция	QCI
$S_1 \wedge S_2$	$P(S_1 \wedge S_2) = K_1^x P(S_1) K_2^x P(S_2)$
$S_1 \vee S_2$	$P(S_1 \vee S_2) = K_1^x P(S_1) + K_2^x P(S_2) - K_1^x P(S_1) K_2^x P(S_2)$
$\neg S$	$P(\neg S) = K_y^x (1 - P(S))$

неопределенных рассуждений. В статье используется вероятностный подход.

QCI запроса равен вероятности P(S). Событие S подробно рассмотрено в предыдущем разделе. QCI вычисляется, исходя из логики запроса. Логические операции "$\wedge$", "$\vee$", "$\neg$" выполняются над независимыми вероятностями гипотез $S_1, S_2, ...S_n$, где n количество атрибутов в Y. Для результирующей формулы вычисления вероятности события S учета вероятностей одних гипотез $S_1, S_2, ...S_n$ не достаточно. Не все условия в запросе однозначно влияют на результат. Если атрибут y_i функционально не определяет X, то условие на этот атрибут при поиске является лишним. «Меру зависимости X от y» задает коэффициент доверия, определяемый экспертом предметной области. $K_y^x \in [0, 1]$, где X — множество имен искомых атрибутов в результирующей схеме отношения, а y — атрибут в Y, с соответствующим ему событием S и гипотезой A. $K_y^x = 1$ означает, что X функционально зависит от атрибута y $(y \to X)$. $K_y^x = 0$ означает, что X функционально не зависит от y. В самом простом случае, когда запрос состоит из одного условия без логических операторов, QCI вычисляется так:

$$QCI = K_1^x P(S_1). \tag{4}$$

С учетом логических операторов QCI вычисляется согласно 1.

Отметим, что по формуле дизъюнкции вероятностей

$$\begin{aligned} P(S_1 \vee S_2) &= K_1^x P(S_1) + K_2^x P(S_2) - P(S_1 \wedge S_2) = \\ &= K_1^x P(S_1) + K_2^x P(S_2) - K_1^x P(S_1) K_2^x P(S_2). \end{aligned}$$

Условия поиска по атрибутам состоят из простых условий вида «имя атрибута» «оператор сравнения» «значение», связанные логическими операторами "$\wedge$", "$\vee$", "$\neg$". В качестве значения может выступать другой атрибут из отношения. Условия поиска могут быть сгруппированы и возможно изменить приоритет одних над другими, что делается с помощью скобок.

Пример, для отношения со схемой $\{a, b, c, d\}$ запрос имеет следующие условия: "$b > 0 \wedge (c = 6 \vee \neg(d =$ "строка$''))$". Тогда, гипотеза S_1 соответствует условию $b > 0$, $S_2 - c = 6$, $S_3 - d =$ "строка$''$. QCI для каждой строки этого запроса вычисляется так:

$$QCI = K_1^x P(S_1)(K_2^x P(S_2) + K_3^x(1 - P(S_3)) - K_2^x P(S_2) K_3^x (1 - P(S_3))).$$

В свою очередь, каждая гипотеза S_i $(1 \leqslant i \leqslant n)$ напрямую зависит от гипотез A_i (атрибут $< a_i,\ t_i >$ существует в схеме) и $\overline{A_i}$ (атрибут $< a_i, t_i >$ не существует в схеме). Для них в свете причинно-следственных отношений S_i — событие. Согласно формуле полной вероятности,

$$P(S_i) = P(A_i)P(S_i|A_i) + P(\overline{A_i})P(S_i|\overline{A_i}) = P(A_i)P(S_i|A_i), \qquad (5)$$

где A_i и $\overline{A_i}$ — полное пространство событий, а $P(S_i|\overline{A_i}) = 0$, по определению гипотезы $\overline{A_i}$.

Для атрибутов, которые не изменились в результирующей схеме отношения по сравнению с начальной схемой, $P(A_i) = 1$. Для атрибутов, которые изменились в результирующей схеме отношения, вычисление $P(A_i)$ задается формулами (1), (2), (3). Для неопределенных атрибутов событие A_i так же неопределенно, поэтому использовать такие атрибуты для поиска Y нельзя. Но событие $\overline{A_i}$ (атрибут не изменился) определено и может входить в Y.

$P(S_i|A_i)$ — «мера достоверности» совершения события S_i при совершении гипотезы A_i. Эту вероятность можно определить экспериментом, результатом которого будет случайная величина η_i.

$$P(S_i|A_i) = \eta_i. \qquad (6)$$

Для вычисления $P(S_i)$ для каждого атрибута в результирующем заголовке Hr_R каждой строки отношения r нужно:

- Определить, степень функциональной зависимости X от каждого y_i в Y. Отразить это значением K_y^x.
- Определить событие S_i и гипотезу A_i для каждого y_i.
- Вычислить последовательно вероятности $P(A_i)$ и $P(S_i|A_i)$, согласно формулам (1), (2), (3), (6).
- Вычислить $P(S_i)$ по формуле (5).

В одном условии поиска может быть задействовано несколько атрибутов. Если в условии требуется сравнить значение двух атрибутов a_1 и a_2, то нужно определять не два разных события $P(S_1)$ и $P(S_2)$, а одно $P(S_{12})$. Это событие определяется гипотезой A_{12} (оба атрибута есть в результирующем отношении), вероятность, которой равна $P(A_{12}) = P(A_1)P(A_2)$, так

как A_1 и A_2 независимы. $P(S_{12})$ определяется коэффициентом доверия, который показывает, как сильно зависит искомый результат сразу от обоих атрибутов. Он может сильно отличатся от коэффициентов доверия для отдельных атрибутов a_1 и a_2.

$$P(S_{12}) = P(A_1)P(A_2)P(S_{12}|A_{12}) = P(A_1)P(A_2)\eta_{12}.$$

Для всего запроса в целом можно также определить степень доверия α, ниже которой результаты QCI не рассматриваются. Аналогично коэффициенту доверия $\alpha \in [0, 1]$.

2.4. Алгоритм поиска в неподдерживаемой БД

Представим алгоритм поиска в неподдерживаемой БД, использующий результирующие схемы отношений.

Вход:

- Результирующая схема отношения Hr_R.
- Анализ предметной области: объекта поиска — множество атрибутов X и его функциональных зависимостей — множества Y.
- Матрица коэффициентов доверия K_y^x для всех возможных пар $< X, y >$, $y \in Y$.
- Поисковый запрос: условия на каждый атрибут из Y, соединенные «$\wedge$», «$\vee$», «$\neg$» с использованием скобок.
- Степень доверия запросу α.

Выход: Отсортированное по атрибуту QCI отношение *search* с заголовком $\{QCI\} \cup X$.

search $= \varnothing$;

Шаг 1. Для каждого атрибута из $Y (1 \leqslant i \leqslant n)$ определить гипотезу A_i и вычислить $P(A_i)$;

Шаг 2. Для каждого кортежа в отношении:

Шаг 2.1. Определить событие S_i для каждого атрибута из $Y (1 \leqslant i \leqslant n)$;

Шаг 2.2. Вычислить вероятности $P(S_i|A_i)$ и $P(S_i)$ для каждого атрибута из $Y (1 \leqslant i \leqslant n)$;

Шаг 2.3. Вычислить QCI, исходя из логики запроса, по всем $P(S_i)$;

Шаг 2.4. Добавление в *search* кортежа из отношения с заголовком $\{QCI\} \cup X$.

Шаг 3. Сортировка всех кортежей *search* по QCI.

Шаг 4. Фильтрация всех кортежей *search* по степени доверия запросу α.

Вычислительная сложность алгоритма зависит от 2-х величин: m — количества кортежей в отношении и n — количества атрибутов во множестве Y. В процессе работы алгоритм считает QCI для всех m строк отношения, рассчитывая для каждой n вероятностей $P(S_i)$, $1 \leqslant i \leqslant n$. Затем, делается сортировка по QCI — самые быстрые алгоритмы, которой имею вычислительную сложность порядка $m \log_2 m$.

Заключение

Представленный алгоритм имеет следующие ограничения:

1. Не трудно заметить, что в алгоритме рассматриваются выборки только в пределах одного отношения. Это ограничение можно обойти средствами используемого в СУБД реляционного языка запросов: представить несколько отношений, как одно и затем применить описанный алгоритм.
2. Алгоритм не решает главных проблем при поиске в БД: проблемы определения функциональных зависимостей. Если эти зависимости определены не верно, то с большой долей вероятности алгоритм будет не эффективен.
3. Использование коэффициентов доверия для определения зависимости результата поиска от его условий является достаточно грубой оценкой. Алгоритм предполагает, что эти коэффициенты определяются экспертом предметной области. Нет никакой гарантии эффективной работы алгоритма, в случае ошибки эксперта. В качестве меры, позволяющей уменьшить влияние эксперта, предполагается ввести некоторый элемент «обучения» в процессе работы. Алгоритм сам меняет начальные коэффициенты доверия в случае необходимости. Качественно проработать такую «обратную связь» мешает слишком сильная зависимость ее от конкретной задачи поиска. Коэффициент доверия отражает лишь зависимость X от атрибута y в «целом» и не отражает зависимость X от конкретных условий на y. Гораздо правильнее считать коэффициент доверия для каждого условия каждого атрибута, по которому ведется поиск. Но хранить и использовать всю эту информацию не целесообразно из-за слишком большого количества разных условий даже в самых простейших задачах поиска.

В статье рассмотрен только алгоритм поиска в «чистой» реляционной модели данных. В промышленных СУБД широко используются неопределенные в реляционной алгебре операции и существуют некоторые отличия, направленные на расширение возможностей и удобство использования хранящихся в СУБД реляционных данных.

Алгоритм был проверен на промышленных базах данных во время опытной эксплуатации системы «Электронный Архив документов индивидуального (персонифицированного) учета» для Пенсионного Фонда России. С помощью него была решена задача односторонней интеграции системы с реляционными БД других систем Пенсионного Фонда.

Литература

1. *Порай Д. С.*, *Тарханов И. А.* Односторонняя интеграция информационных систем // Информационно-аналитические аспекты в задачах управления: Труды ИСА РАН / Под ред. В. Л. Арлазарова и Н. Е. Емельянова. М.: Издательство ЛКИ/URSS, 2007.
2. *Дейт К.*, *Дарвен Х.* Основы будущих систем баз данных. Третий манифест. М: Янус-К, 2004.
3. *Дейт К.* Введение в системы баз данных: 7-е изд. М., СПб.: Вильямс, 2000.
4. *Кузнецов С. Д.* Базы данных. Языки и модели. М.: Бином Пресс, 2008.
5. *Пушников А. Ю.* Введение в системы управления базами данных. Ч. 1. Реляционная модель данных: Учебное пособие. Изд-е Башкирского ун-та. Уфа, 1999. 108 с.
6. *Дюбуа Д.*, *Прад Г.* Теория возможностей. Приложения к представлению знаний в информатике М.: Радио и связь, 1990.
7. *Dubois D.*, *Prade H.* Using Fuzzy Sets in Database Systems: Why and How? // Proc. of 1996 Workshop on Flexible Query-Answering systems (FQAS'96), Denmark, May 22–24, 1996, P. 89–103.
8. *Рассел С.*, *Норвиг. П.* Искусственный интеллект: современный подход: 2-е изд. М., СПб.: Вильямс, 2006.