Анализ зависимостей соединения в базах данных

С.В. Зыкин

Институт Математики им. С.Л. Соболева СО РАН (ОФ)

Введение

- Пусть задана реляционная база данных: (R_1, R_2, ..., R_k), полученная в результате синтеза отношений R_i, [R_i] – совокупность атрибутов (схема отношения R_i), $R_i[V]$ – проекция отношения R_i на множество атрибутов V, $U=\{A_1, A_2, ..., A_n\}$ – конечное множество всех атрибутов, на которых заданы отношения БД.
- **Определение 1**. Зависимость соединения $\bowtie(X_1, X_2,\ldots, X_k)$ выполнима (implies), если для любой реализации отношения R выполнено:

$$R[X]=R[X_1]\bowtie R[X_2]\bowtie\ldots\bowtie R[X_k], \qquad (1)$$

где $\bowtie$ – операция естественного соединения [Ullman80], $X=X_1\cup X_2\cup\ldots\cup X_k$.

Введение

- В работе [Sciore82] используется понятие полной (full) зависимости соединения. Пусть $S=\{X_1, X_2,\ldots, X_k\}$ схема БД, удовлетворяющая условию полноты, тогда $\bowtie(X_1, X_2,\ldots, X_k)$ называется полной зависимостью соединения. При этом, схема БД S называется полной, если $attr(S)=U$, где $attr(S)=X_1\cup X_2\cup\ldots\cup X_k$.
- В работе [Zykin21] рассмотрено понятие области определения зависимости соединения:
- **Определение 2**. Областью определения зависимости соединения $\bowtie(X_1, X_2,\ldots, X_k)$ в отношения R будем называть множество $X=X_1\cup X_2\cup\ldots\cup X_k\subseteq U$.
- Таким образом, зависимость $\bowtie(X_1, X_2,\ldots, X_k)$ является полной в проекции $R[X]$, где $X=X_1\cup X_2\cup\ldots\cup X_k$, и в общем случае она является встроенной в отношении R.

Прогонка

- **Определение 3**. Правило $\bowtie(X_1, X_2,\ldots, X_k),\ldots, \bowtie(Z_1, Z_2,\ldots, Z_n) \models \bowtie(Y_1, Y_2,\ldots, Y_m)$ выполнимо, если для любой реализации отношения R, удовлетворяющей зависимостям соединения $\bowtie(X_1, X_2,\ldots, X_k),\ldots, \bowtie(Z_1, Z_2,\ldots, Z_n)$, имеет место равенство:

$$R[Y]=R[Y_1]\bowtie R[Y_2]\bowtie\ldots\bowtie R[Y_m],$$

где $Y=Y_1\cup Y_2\cup\ldots\cup Y_m$.

- Для определения выполнимости зависимости $\bowtie(Y_1, Y_2,\ldots, Y_m)$ традиционно используется вычислительный метод, называемый прогонкой (chase) [Maier83]. Исходными данными для метода являются зависимости соединения и функциональные зависимости, которым должна удовлетворять любая реализация отношения R.

Сравнение подходов

- Рассмотрим пример: проверим выполнимость правила $\bowtie(AB, BC) \models \bowtie(ABD, BCD)$, где A, B, C, D – отдельные атрибуты. Начальное табло для зависимости $\bowtie(ABD, BCD)$ представлено в таблице.

A	B	C	D
a_1	a_2	b_1	a_4
b_2	a_2	a_3	a_4

- С использованием метода, предложенного в данной работе, был получен контрпример (кортеж (2, 1, 2, 1)):

A	B	C	D
2	1	2	2
1	1	2	1
2	1	1	1
1	1	1	1

Связанные работы

- Наиболее близка к рассматриваемой тематике работа [2008_Hartmann], в которой используются табло прогонки и аналитическое табло (the Chase and analytical tableau) в качестве инструмента автоматического создания реализаций баз данных, удовлетворяющих значительному количеству ограничений целостности.
- В работе [2016_Xiaocheng] представлено исследование алгоритмов проверки выполнимости зависимостей соединения с точки зрения количества операций ввода-вывода во внешнюю память компьютера. Доказана NP-сложность проверки. Представлен эффективный с точки зрения ввода-вывода алгоритм проверки существования нетривиальной зависимости соединения, удовлетворяющей текущему состоянию базы данных.

Организация тестирования

- Генерируемое отношение в общем виде представлено в таблице

$R =$

A_1	A_2	A_3	…	A_n
a_{11}	a_{12}	a_{13}	…	a_{1n}
a_{21}	a_{22}	a_{23}	…	a_{2n}
…	…	…	…	…
a_{m1}	a_{m2}	a_{m3}	…	a_{mn}

- Символом n в таблице 3 обозначена арность отношения R (количество столбцов), m - кардинальность отношения (количество кортежей), A_j – имя атрибута, a_{ij} – значения атрибута A_j. Допустим, что каждое значение a_{ij} является целочисленным в интервале от 1 до l. Тогда общее количество реализаций отношения R равно l^{nm}.

Анализ процесса тестирования

- Наибольший вклад в количество генерируемых отношений вносит интервал значений a_{ij}. Если учитывать наличие неопределенных значений *Null* для некоторых атрибутов в реальных базах данных. Результат сравнения этого значения, в том числе и с другим значением *Null*, в команде SQL получит значение *unknown*: «0≠0».
- На следующем этапе необходимо выполнить сокращение количества кортежей в реализации отношения *R*. Это возможно за счет удаления (игнорирования) дублированных кортежей. Схема генерации представлений *R* должна исключать появление дублированных кортежей, чтобы исключить последующее манипулирование ими при проверке выполнимости всех зависимостей Σ и σ на текущей реализации отношения *R*.

Использование свойств зависимостей

В работе [Zykin21] представлено множество из пяти правил:

•P0) $\emptyset \models \bowtie(X)$, $X \subseteq U$.

•P1) $\bowtie(X_1, X_2, \ldots, X_k) \models \bowtie(Y_1, Y_2, \ldots, Y_l)$, если:

- p11) для любого X_i существует Y_j: $X_i \subseteq Y_j$;
- p12) для любого Y_j выполнено $Y_j \subseteq X_1 \cup X_2 \cup \ldots \cup X_k$.

•P2) $\bowtie(X_1, X_2, \ldots, X_k) \models \bowtie(Y_1, Y_2, \ldots, Y_k)$, если:

- p21) $X_i \cap (X_1 \cup X_2 \cup \ldots \cup X_{i-1} \cup X_{i+1} \ldots \cup X_k) \subseteq Y_i$;
- p22) $Y_i \subseteq X_i$.

•P3) $\bowtie(X_1, X_2, \ldots, X_k) \models \bowtie(Y_1, Y_2, \ldots, Y_k)$, если:

- p31) $Y_i = X_i \cup Z$, $Z \subseteq X_1 \cup X_2 \cup \ldots \cup X_k$.

•P4) $\bowtie(X_1, X_2, \ldots, X_k, V)$, $\bowtie(Y_1, Y_2, \ldots, Y_l) \models \bowtie(X_1, X_2, \ldots, X_k, V \cap Y_1, V \cap Y_2, \ldots, V \cap Y_l)$, если:

- p41) $Y_i \cap Y_j \subseteq V$, $i \neq j$;
- p42) $V \cap (X_1 \cup X_2 \cup \ldots \cup X_k) \subseteq V \cap (Y_1 \cup Y_2 \cup \ldots \cup Y_l)$.

Редукция зависимости соединения

•**Определение 4**. Зависимость соединения $\bowtie(Y_1, Y_2,\ldots, Y_l)$ будем называть редукцией зависимости $\bowtie(X_1, X_2,\ldots, X_k)$, если для любого X_i существует Y_j: $X_i \subseteq Y_j$, для любого Y_j существует X_i: $Y_j \subseteq X_i$, и не существует Y_s и Y_p: $Y_s \subseteq Y_p$, $s \neq p$.

•Зависимость соединения будет редуцирована, если из нее удалены компоненты, являющиеся подмножеством других компонентов в этой зависимости.

•**Утверждение 1**. *Произвольная зависимость соединения и ее редукция эквивалентны.*

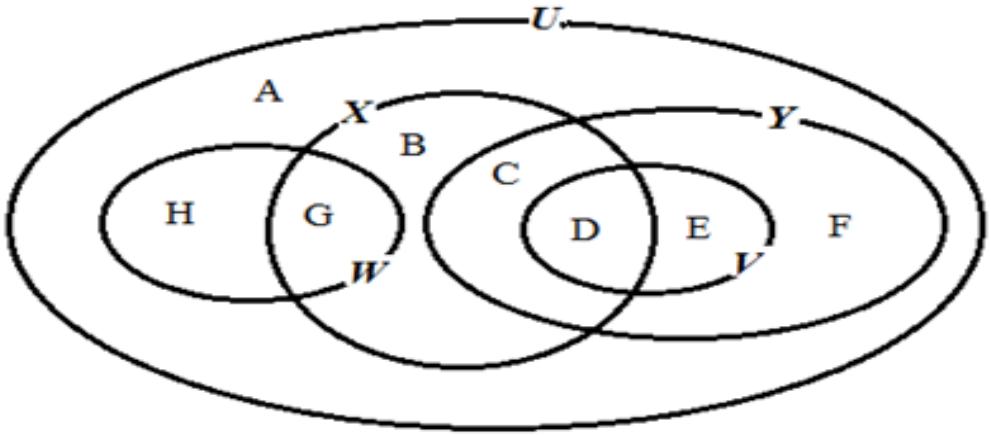
•Замечание. При построении редукции нельзя удалять атрибуты, используемые только в одном X_i, как это делается в алгоритме Грэхема при проверке ацикличности гиперграфа. Поскольку эти зависимости не будут эквивалентны исходным зависимостям.

Выбор множества атрибутов

•Проведем анализ аксиомы A8 многозначных зависимостей. Обобщением данной аксиомы является правило:

$$\bowtie(XY, X(U–Y)), \bowtie(WV, W(U–V)) \models \bowtie(XV, X(U–V)), \quad (2)$$

где U – множество всех атрибутов (аксиома определена только для полных зависимостей). На рисунке представлены все возможные пересечения множеств атрибутов для аксиомы A8:



$$\bowtie(BCDEFG, ABCDGH), \bowtie(DEGH, ABCFGH) \models \bowtie(BCDEG, ABCDFG) \quad (3)$$

Признак зависимости соединения

•**Утверждение 2**. *Зависимость* $\bowtie(X_1, X_2,\ldots, X_k)$ *выполнена в R, если для любой реализации R наличие кортежей* $t_1, t_2,\ldots, t_k$, *не обязательно различных, гарантирует наличие кортежа* $t\in R$: $t[X_i]=t_i[X_i]$, $i=1, 2,\ldots, k$, *если* $t_i[X_i\cap X_j]=t_j[X_i\cap X_j]$ *когда* $X_i\cap X_j\neq\varnothing$.

•Следствие. Если не учитывать значение *Null*, то из доказательства утверждения 2 следует, что при тестировании зависимости $\bowtie(X_1, X_2,\ldots, X_k)$ достаточно проверять условие $R[X_1]\bowtie R[X_2]\bowtie\ldots\ \bowtie R[X_k]\subseteq R[X]$, то есть каждый кортеж соединения проекций $R[X_i]$ должен принадлежать проекции $R[X]$. Кроме того, для проверки на выполнимость зависимости достаточно формировать не менее $k+1$ различных кортежей, где k количество компонентов в зависимости.

Признак завершения тестирования

•Пусть $R_X = R[X_1] \bowtie R[X_2] \bowtie \ldots \bowtie R[X_k]$, $X = X_1 \cup X_2 \cup \ldots \cup X_k$, $R_Y = R[Y_1] \bowtie R[Y_2] \bowtie \ldots \bowtie R[Y_l]$, и $Y = Y_1 \cup Y_2 \cup \ldots \cup Y_l$.

•**Утверждение 3**. *Правило* $\bowtie(X_1, X_2, \ldots, X_k) \models \bowtie (Y_1, Y_2, \ldots, Y_l)$ *выполнимо в R при* $Y=X$, *если для любой реализации R выполнено*: $|R_X| \geq |R_Y|$, *где* $|R_X|$ – *кардинальность* R_X.

•Замечание. Если в процессе моделирования по истечении приемлемого времени ни разу не было нарушено условие $|R_X| \geq |R_Y|$, то велика вероятность отсутствия контрпримера и можно приступать к доказательству надежности соответствующего правила вывода. Рассмотренное условие применимо только к правилам с одной зависимостью в левой части.

Схема тестирования правил

•Вход: тестируемое правило и количество кортежей в реализации k+1.

•**Шаг 1**. Генерация очередной реализации отношения R. Если очередная реализация не существует, то контрпримера нет и конец алгоритма.

•**Шаг 2**. Дополнение недостающих кортежей из R_X в конец отношения R, где $R_X = R[X_1] \bowtie R[X_2] \bowtie \ldots \bowtie R[X_k]$ и $X = X_1 \cup X_2 \cup \ldots \cup X_k$.

•**Шаг 3**. Проверка выполнимости оставшихся зависимостей в левой части правила на реализации R. Если существует невыполнимая зависимость, то переход на шаг 1.

•**Шаг 4**. Проверка зависимости в правой части правила. Если зависимость невыполнима, то контрпример R найден и конец алгоритма. Иначе переход на шаг 1.

Алгоритм генерации отношений

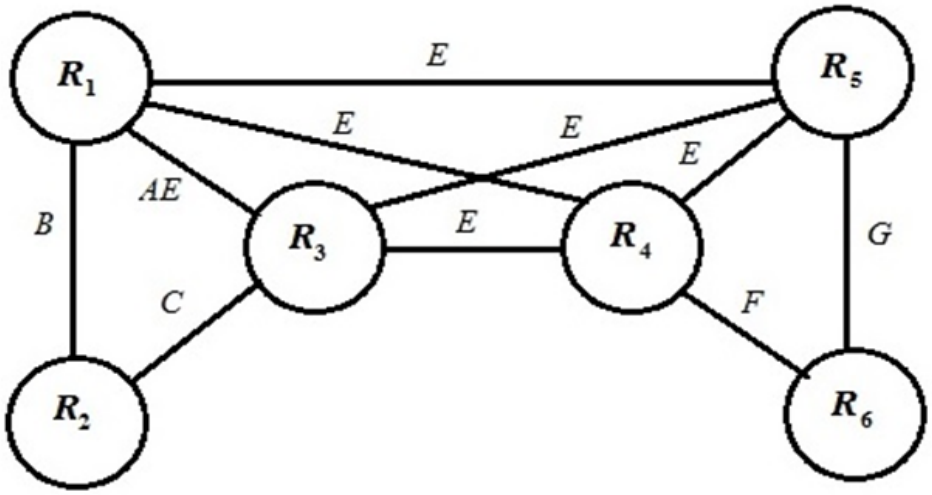
```
SUB Gener(num_t, n_col, n_tup, t, min_a, max_a)
IF num_t=0 THEN
    DO i=1 TO n_col
        t(1,j)=min_a
        t(2,j)=min_a
    END DO
    num_t=2
END IF
DO WHILE num_t>0
    IF Next(num_t, n_col, t, min_a, max_a) THEN
        IF num_t=n_tup THEN
            EXIT SUB
        ELSE
            num_t=num_t+1
            DO i=1 TO n_col
                t(num_t,j)=t(num_t–1,j)
            END DO
        END IF
    ELSE
        num_t=num_t–1
    END IF
END DO
END SUB
```

Алгоритм генерации кортежа

```
FUNC Next(num_t, n_col, t, min_a, max_a)
i=n_col
DO WHILE i>0
    IF t(num_t, i)<max_a THEN
        t(num_t, i)=t(num_t, i)+1
        Next=TRUE
        EXIT FUNC
    ELSE
        IF i>1 THEN
            DO j=i TO n_col
                t(num_t,j)=min_a
            END DO
            i=i–1
        ELSE
            Next=FALSE
            EXIT FUNC
        END IF
    END IF
END DO
END FUNC
```

Анализ результатов тестирования

- Для тестирования правил используем зависимость:
  - $\bowtie(ABE, BC, ACE, EF, EG, FG)$.
- Структура связей зависимости соединения:



- На рисунке представлены отношения базы данных: $R_1(ABE)$, $R_2(BC)$, $R_3(ACE)$, $R_4(DEF)$, $R_5(EG)$, $R_6(FG)$. Ребрами на рисунке обозначены общие атрибуты в отношениях, а узлами являются отношения R_i.

Анализ результатов тестирования

•Далее будем исследовать правила следующего вида:

$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(Y_1, Y_2),$$

где Y_1 и Y_2 различные компоненты зависимости.

•Выполнимы только следующие правила:

$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(ABE, DEF),$$
$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(ABE, EG),$$
$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(ACE, DEF),$$
$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(ACE, EG).$$

•Для всех остальных правил с двумя компонентами в правой части был найден контрпример.

•Объединение левых частей двух выполнимых правил оказалось невыполнимо:

$$\bowtie(ABE, BC, ACE, EF, EG, FG) \models \bowtie(ABE, DEF, EG)$$

Тестирование суперключа

•Пример проектирования фрагмента БД: аренда изделий различного типа, у типа изделия несколько собственников.

• Рассмотрим минимальный набор атрибутов:

A_1 – *инвентарный номер изделия*,

A_2 – *дата выдачи изделия*,

A_3 – *идентификатор клиента*,

A_4 – *плановая дата возврата изделия*,

A_5 – *номер собственника изделия*,

A_6 – *название собственника изделия*,

A_7 – *номер типа изделия*,

A_8 – *название изделия*,

A_9 – *характеристика изделия*.

•$F=\{A_1A_2A_3 \rightarrow A_4,\ A_5 \rightarrow A_6,\ A_1 \rightarrow A_7,\ A_7 \rightarrow A_8A_9\}$.

Начальная схема БД

Выдача изделий в прокат (R_1)

инвентарный номер изделия	**дата выдачи изделия**	**идентификатор клиента**	плановая дата возврата изделия

Владельцы изделий (R_2)

номер собственника изделия	название собственника изделия

Инвентаризация изделий (R_3)

инвентарный номер изделия	номер типа изделия

Описание изделий (R_4)

номер типа изделия	название изделия	характеристика изделия

Анализ суперключа

•Суперключ:

инвентарный номер изделия	дата выдачи изделия	идентифика-тор клиента	номер собственни-ка изделия

В этом отношении присутствует аномалия дополнения-модификации: номера собственников изделия надо повторять столько раз, сколько в БД зарегистрировано изделий данного типа и сколько раз каждое изделие выдавалось в прокат.

Имеет место МЗ: $A_1 \twoheadrightarrow A_2 A_3 | A_5$

инвентарный номер изделия	дата выдачи изделия	идентифика-тор клиента

и

инвентарный номер изделия	номер собственни-ка изделия

Анализ суперключа

•Первое отношение часть R_1. Во втором есть аномалия. Зависимость $A_1 \twoheadrightarrow A_2A_3 | A_5$ эквивалентна зависимости соединения $\bowtie(A_1A_2A_3,\ A_1A_5)$. Рассмотрим правило (не выводимо из P0–P4):

$$\bowtie(A_1A_2A_3, A_1A_7, A_5A_7) \vDash \bowtie(A_1A_2A_3, A_1A_5).$$

• Тестирование этого правила не выявило контрпримера. Следовательно, зависимость $\bowtie(A_1A_2A_3,\ A_1A_5)$ выводима. Сделаем декомпозицию в соответствии с правилом. Итоговая схема БД дополнится только одним отношением:

Собственность (R_5)

номер типа изделия	номер собственника изделия

•Отношения R_1 – R_5 находятся в пятой нормальной форме и не содержит ни одной аномалии.

Благодарю за внимание!

Вопросы?