Базы данных
функциональные зависимости

В.Н.Лукин

11.12.2020

Функциональная зависимость

Теория нормальных форм базируется на понятии функциональных зависимостей (ФЗ).

Мы знаем, что функциональная зависимость имеет место, если значение кортежа на одном множестве атрибутов однозначно определяет его значение на другом.

Дадим определение функциональной зависимости на некотором отношении, а затем рассмотрим функциональные зависимости на схеме.

.

Функциональная зависимость
определение

Пусть r — отношение со схемой R, $X,Y \subseteq R$. Отношение r удовлетворяет функциональной зависимости $X \to Y$, если $\pi_Y(\sigma_{X = x}(R))$ имеет не более, чем один кортеж для каждого X-значения x, то есть для $t_1, t_2 \in r$: $(t_1(X)=t_2(X)) \Rightarrow (t_1(Y) = t_2(Y))$.

Очевидно, что $X \to \varnothing$ тривиально удовлетворяет любому отношению, а $\varnothing \to Y$ удовлетворяет отношениям, в которых *Y*-значения всех кортежей совпадают.

.

Функциональная зависимость
алгоритм нахождения ФЗ

Из определения, для отношения r и множеств атрибутов $X, Y \subseteq R$ строится алгоритм (*SATISFIES*), проверяющий существование функциональной зависимости $X \to Y$. Его суть: кортежи группируются по X-значениям, при этом все Y-значения для одинаковых X-значений должны быть одинаковыми.

Алгоритм

Вход: отношение r, функциональная зависимость $X \to Y$.

Выход: **истина**, если r удовлетворяет $X \to Y$, **ложь** в противном случае.

Отсортировать r по X-столбцам, группируя равные значения.

Если каждая группа X-кортежей имеет одинаковые Y-значения, возвратить **истину**, в противном случае — **ложь**.

Функциональная зависимость
пример (1)

Задан график распределения пилотов по рейсам. Проверим, удовлетворяет ли он функциональным зависимостям *Рейс* $\rightarrow$ *Время* и *Время* $\rightarrow$ *Рейс.*

Пилот	***Рейс***	***Дата***	***Время***
Иванов	83	09.01.2020	10:00
Иванов	16	10.01.2020	13:00
Петров	81	08.01.2020	05:00
Петров	31	12.01.2020	18:00
Петров	83	11.01.2020	10:00
Сидоров	83	13.01.2020	10:00
Сидоров	16	12.01.2020	13:00
Федоров	81	09.01.2020	05:00
Федоров	81	13.01.2020	05:00

Функциональная зависимость
пример (2)

Легко видеть, что первая функциональная зависимость удовлетворяется, вторая — нет.

Пилот	***Рейс***	***Дата***	***Время***
Иванов	83	09.01.2020	10:00
Петров	83	11.01.2020	10:00
Сидоров	83	13.01.2020	10:00
Петров	81	08.01.2020	05:00
Федоров	81	09.01.2020	05:00
Федоров	81	13.01.2020	05:00
Федоров	41	15.01.2020	13:00
Петров	31	12.01.2020	18:00
Иванов	16	10.01.2020	13:00

Пилот	***Рейс***	***Дата***	***Время***
Иванов	83	09.01.2020	10:00
Петров	83	11.01.2020	10:00
Сидоров	83	13.01.2020	10:00
Петров	81	08.01.2020	05:00
Федоров	81	09.01.2020	05:00
Федоров	81	13.01.2020	05:00
Федоров	41	15.01.2020	13:00
Петров	31	12.01.2020	18:00
Иванов	16	10.01.2020	13:00

Аксиомы вывода (1)

Для отношения *r*(*R*) всегда существует множество ФЗ, которому оно удовлетворяет, но одно его состояние может удовлетворять данной ФЗ, другое — нет. Требуется выявить семейство ФЗ *F*, которому удовлетворяют все допустимые состояния *r*, (определить семейство *F*, заданное на схеме *R*).

Множество ФЗ для *r*(*R*), конечно, поэтому можно найти все ФЗ, которым удовлетворяет *r*. Но это долгий процесс, который зависит от мощности множества ФЗ и от количества атрибутов в этих зависимостях.

Аксиомы вывода (2)

Вопрос, можно ли для данного множества ФЗ найти семантически эквивалентное, но меньшее множество? Иногда это возможно.

Определение. *Будем говорить, что множество ФЗ F влечет ФЗ* $X \rightarrow Y$ ($F \vDash X \rightarrow Y$)*, если каждое отношение, удовлетворяющее всем зависимостям из F, удовлетворяет и* $X \rightarrow Y$.

Аксиомы Армстронга (1)

В 1974 году В. Армстронг предложил правила, пользуясь которыми можно по множеству функциональных зависимостей F построить множество G такое, что если отношение удовлетворяет всем $f \in F$, оно удовлетворяет и всем $g \in G$.

Эти правила, устанавливающие определенные свойства функциональных зависимостей, названы аксиомами вывода.

Аксиомы Армстронга (1)

*F*1. *Рефлективность*:

$X \to X$

*F*2. *Пополнение* (*расширение левой части*):

$(X \to Y) \Rightarrow (XZ \to Y)$

Пример 1. R(A, B, C, D)

A	B	C	D
$a1$	$b1$	$c1$	$d1$
$a2$	$b2$	$c1$	$d1$
$a1$	$b1$	$c1$	$d2$
$a3$	$b3$	$c2$	$d3$

$(A \to B) \Rightarrow \{ AB \to B, AC \to B, AD \to B, ABC \to B, ABD \to B \}$

Аксиомы Армстронга (2)

$F3$. *Аддитивность*.

Объединяются две ФЗ с одинаковыми левыми частями:

$$(X \to Y, X \to Z) \Rightarrow (X \to YZ).$$

Пример 2. Для отношения из примера 1:

$$(A \to B, A \to C) \Rightarrow (A \to BC).$$

$F4$. *Проективность*.

$$(X \to YZ) \Rightarrow (X \to Y).$$

$F5$. *Транзитивность*:

$$(X \to Y, Y \to Z) \Rightarrow (X \to Z).$$

Пример 3. Для отношения из примера 1:

$$(A \to B, B \to C) \Rightarrow (A \to C).$$

Аксиомы Армстронга (3)

*F*6. *Псевдотранзитивность*:

$$(X \to Y,\ YW \to Z) \Rightarrow (XW \to Z).$$

Эта система аксиом избыточна: очевидно, что $F6 \Rightarrow F5$ (для $W = \varnothing$), $(F1, F2, F3, F5) \Rightarrow F6$. Но она полна, т. е. любая ФЗ, которая следует из *F*, может быть получена применением аксиом *F*1–*F*6.

Доказыватся, что {*F*1, *F*2, *F*6} — полное подмножество аксиом: $(F1, F2, F6) \Rightarrow F3$, $(F1, F2, F6) \Rightarrow F4$, $F6 \Rightarrow F5$.

Докажем $(F1, F2, F6) \Rightarrow F4$. Пусть $X \to YZ$, тогда из (*F*1):

$Y \to Y$ и (F2): $YZ \to$ Y. По (*F*6): $X \to Y$. Доказано.

Подмножество аксиом {*F*1, *F*2, *F*6} назвается *аксиомами Армстронга*.

Аксиомы Армстронга (4)

Определение. *Пусть F — множество ФЗ для $r(R)$. Замыкание F (обозначается F^+) — это наименьшее множество, содержащее F, и такое, что при применении к нему аксиом Армстронга нельзя получить ни одной ФЗ, не принадлежащей F^+.*

Определение. *Два множества ФЗ F и G над одной и той же схемой называются эквивалентными, $F \equiv G$, если $F^+ = G^+$.*

Определение. *Используемое множество в последовательности вывода P — множество ФЗ из F, принадлежащее P.*

Аксиомы Армстронга (6)

***Пример* 4.**

$$F = \{AB \to C,\ BC \to D,\ C \to E,\ AE \to F,\ DE \to G\}.$$

Последовательность вывода для $AB \to EG$ может выглядеть, как показано на следующем слайде.

Справа указана аксиома и номера элементов последовательности, к которым она применяется.

Очевидно, что эта последовательность будет, в частности, включать последовательность вывода для некоторых других ФЗ, например, $AB \to DE$.

Аксиомы Армстронга (7)

1. $AB \to C$;
2. $AB \to AB$ (*F*1: 1);
3. $AB \to B$ (*F*4: 2);
4. $AB \to BC$ (*F*3: 1, 3);
5. $BC \to D$;
6. $AB \to D$ (*F*5: 4, 5);
7. $C \to E$;
8. $AB \to E$ (*F*5:1, 7);
9. $AB \to DE$ (*F*3: 6, 8);
10. $DE \to G$;
11. $AB \to G$ (*F*5: 9, 10);
12. $AB \to EG$ (*F*3: 8, 11).

B-аксиомы (1)

*B*1. *Рефлективность* (*Reflexivity*)

$$X \to X$$

*B*2. *Накопление* (*Accumulation*)

$$(X \to YZ, Z \to CW) \Rightarrow (X \to YZCW)$$

*B*3. *Проективность* (*Projectivity*)

$$(X \to YZ) \Rightarrow (X \to Y)$$

Можно показать, что аксиомы Армстронга выводятся из *B*-аксиом. Из полноты системы аксиом Армстронга следует и полнота системы *B*-аксиом.

***Пример* 5.** Пусть F — множество ФЗ из примера 4 (сл.14). Приведем последовательность вывода для $AB \to EG$, использующую только *B*-аксиомы.

В-аксиомы (2)

1. $CD \to CD$ $(B1)$;
2. $C \to E$;
3. $CD \to CDE$ $(B2)$;
4. $CD \to DE$ $(B3)$;
5. $DE \to G$;
6. $CD \to DEG$ $(B2)$;
7. $CD \to EG$ $(B3)$;
8. $AB \to AB$ $(B1)$;
9. $AB \to C$;
10. $AB \to ABC$ $(B2)$;
11. $BC \to D$;
12. $AB \to ABCD$ $(B2)$;
13. $AB \to ABCDE$ $(B2)$;
14. $AB \to ABCDEG$ $(B2)$;
15. $AB \to EG$ $(B3)$.

RAP-последовательности (1)

***Определение*.** Последовательность вывода $X \to Y$ из F, полученная при помощи B-аксиом, называется RAP-последовательностью, если она удовлетворяет следующим условиям:

- первая функциональная зависимость — это $X \to X$;
- последняя функциональная зависимость — это $X \to Y$;
- каждая из остальных функциональных зависимостей либо принадлежит F, либо имеет вид $X \to Z$ и получена с помощью аксиомы B2.

***Теорема*.** Пусть F — множество ФЗ. Если существует последовательность вывода из F для $X \to Y$, то существует RAP-последовательность вывода из F для $X \to Y$

RAP-последовательности (2)

Пример 6. Пусть F — множество ФЗ из примера 4. Выпишем RAP-последовательность вывода $AB \to EG$ из F.

1. $AB \to AB$ $(B1)$;
2. $AB \to C$;
3. $AB \to ABC$ $(B2)$;
4. $BC \to D$;
5. $AB \to ABCD$ $(B2)$;
6. $C \to E$;
7. $AB \to ABCDE$ $(B2)$;
8. $DE \to G$;
9. $AB \to ABCDEG$ $(B2)$;
10. $AB \to EG$.

Ориентированный ациклический граф вывода (1)

Ориентированный (*directed*) ациклический (*acyclic*) граф (*DA*-граф) — орграф без циклов. В помеченном *DA*-графе каждой вершине поставлен в соответствие некоторый элемент из множества меток *L.*

***Определение*.** Пусть F — множество ФЗ над схемой R. DA-граф вывода над F — это DA-граф, помеченный символами атрибутов из R и построенный по следующим правилам:

1. Множество изолированных вершин является DA-графом вывода над F.
2. Пусть DA-граф вывода над F содержит **n** вершин ν_i с метками A_i и в F существует ФЗ $A_1A_2…A_k \rightarrow CZ$ ($k \leq n$). Определим новый граф, добавив к исходному DA-графу вывода вершину ν с меткой C и дуги (ν_1, ν), (ν_2, ν),…, (ν_k, ν). Полученный граф является DA-графом вывода над F.
3. Никакой другой граф не является DA-графом вывода над F.

Ориентированный ациклический граф вывода (2)

DA-граф вывода над *F* называют *DDA*-графом над *F* (*derivation directed acyclic*). Любой *DDA*-граф получается однократным применением правила (1) и многократным применением правила (2).

***Определение*.** Вершина DDA-графа, которая не имеет входящих дуг, называется начальной. Начальная вершина добавляется с помощью правила (1).

***Определение*.** DDA-граф H называется DDA-графом для X $\rightarrow$ Y, если X — множество меток начальных вершин и каждый атрибут в Y — метка вершины в H.

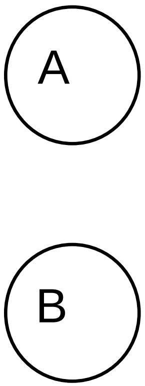
***Определение*.** Используемым множеством **U**(H) в DDA-графе H над F называется множество всех ФЗ в F, использованных при применении правила (2) во время построения графа H.

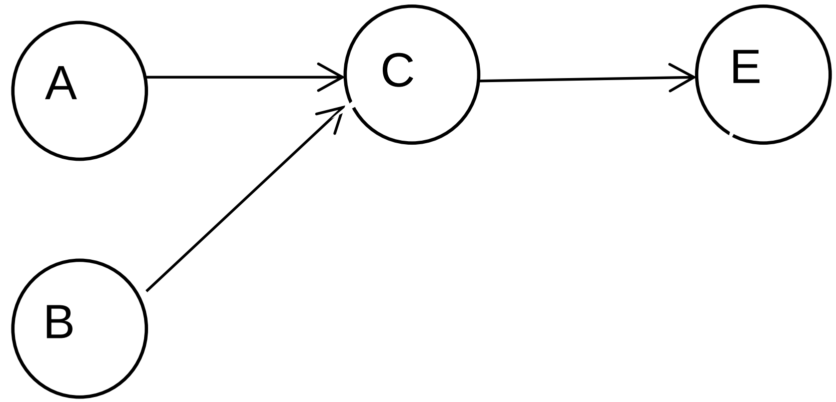
Ориентированный ациклический граф вывода (3)

Пример 7. Пусть F — множество ФЗ из примера 4: $F = \{AB \to C, BC \to D, C \to E, AE \to F, DE \to G\}$. Представим этапы построения DDA-графа над F для $AB \to EG$.

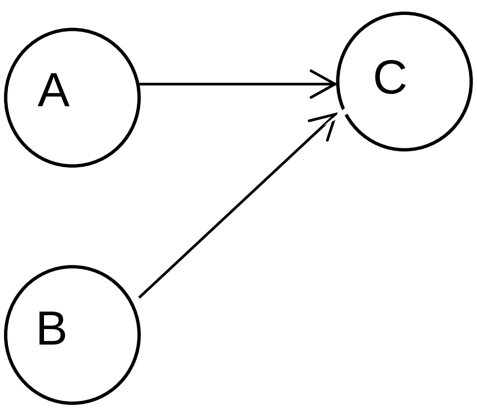
Здесь DDA-граф для ФЗ $AB \to AB$ содержит две изолированные вершины. Его используемое множество пусто.

Ориентированный ациклический граф вывода (4)

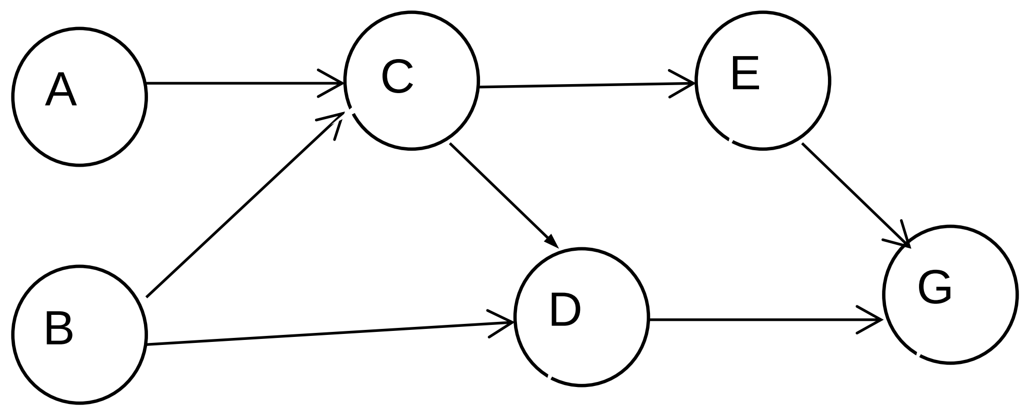




C→E



AB→C



BC→D и DE→G

Ориентированный ациклический граф вывода (5)

***Теорема*.** Для множества ФЗ F над R и ФЗ X $\rightarrow$ Y следующие утверждения эквивалентны.

- F |= X $\rightarrow$ Y.
- Существует последовательность вывода на F для X $\rightarrow$ Y.
- Существует DDA-граф над F для X $\rightarrow$ Y.

***Следствие*.** DDA-граф H над F для X $\rightarrow$ Y с U(H) **=** G существует тогда и только тогда, когда существует RAP-последовательность вывода на F для X $\rightarrow$ Y с используемым множеством G.

При построении множества ФЗ возникает вопрос: выводима ли добавляемая ФЗ из существующих? Если это так, она лишняя. То есть, надо проверить, содержится ли она в замыкании множества ранее существующих зависимостей. Существует алгоритм, проверяющий принадлежность $X \rightarrow Y$ множеству F^+ с временной сложностью $O(n)$, где n — количество ФЗ в F.

Реляционная база данных (1)

Ключ для схемы R — это подмножество $K \subseteq R$, такое, что любое отношение $r(R)$ удовлетворяет ФЗ $K \rightarrow R$, но никакое собственное подмножество $K' \subset K$ этим свойством не обладает.

Схема R отношения состоит из двух частей: S и $\boldsymbol{K}$, где S — множество атрибутов, $\boldsymbol{K}$ — множество выделенных ключей. Введем обозначение: $R = (S, \boldsymbol{K})$.

Определение. *Схемой реляционной БД* $\mathbf{R}$ *над множеством атрибутов* $\mathbf{U}$ *называется совокупность схем отношений* $\{R_1, R_2, \ldots, R_n\}$, *где* $R_i = (S_i, \boldsymbol{K_i})$, $i = 1,2,\ldots, n$, $\cup^n{}_{i=1} S_i = \boldsymbol{U}$, $S_i \neq S_j$ *при* $i \neq j$.

Согласно этому определению, в базе данных не может быть двух отношений с одинаковыми схемами.

Реляционная база данных (2)

***Определение*.** Реляционной БД d со схемой БД $\boldsymbol{R}$ называется такая совокупность отношений $d=\{r_1, r_2,\ldots, r_n\}$, $r_i = r_i(S_i)$, $i=1,\ldots,n$, что для каждой схемы $R_i = (S_i, \boldsymbol{K}_i) \in \boldsymbol{R}$ существует отношение $r_i(S_i) \in d$, удовлетворяющее каждому ключу из $\boldsymbol{K}_i$.

Согласно этому определению, в базе данных не может быть двух отношений с одинаковыми схемами.

Реляционная база данных (пример)

Расписание (*Пилот, Рейс, Дата*) **Время** (*Рейс, Время*)

Иванов	83	09.11.20
Петров	83	11.11.20
Сидоров	83	13.11.20
Петров	81	08.11.20
Федоров	81	09.11.20
Федоров	81	13.11.20
Федоров	41	15.11.20
Петров	31	12.11.20
Иванов	16	10.11.20
Сидоров	16	12.11.20

83	10:00
81	05:00
41	13:00
31	18:00
16	13:00

База данных *d* = {***расписание***, ***время***} имеет схему

R = {(*Пилот Рейс Дата*, {*Пилот Дата*}), (*Рейс Время*, {*Рейс*})}.

Представление множества ФЗ (1)

***Определение*.** Если K — выделенный ключ из $\boldsymbol{K}$, схема $R = (S, \boldsymbol{K})$ включает ФЗ $K \to R$.

***Определение*.** Схема БД $\boldsymbol{R} = \{R_1, R_2, \ldots, R_n\}$ представляет множество ФЗ $G = \{K \to Y \mid \exists R_i \in \boldsymbol{R}, K \to Y \in R_i\}$. Схема БД $\boldsymbol{R}$ полностью характеризует множество ФЗ F, если $F \equiv G$.

Пример.

Схема БД из предыдущего примера представляет множество ФЗ

$G = \{$*Пилот Дата* $\to$ *Рейс Дата, Рейс* $\to$ *Рейс Время*$\}$.

Она полностью характеризует множество

$F = \{$*Пилот Дата* $\to$ *Рейс Время, Рейс* $\to$ *Время*$\}$.

Представление множества ФЗ (2)

Определение. ФЗ $X \to Y$ применима к R, если $X \subseteq R$ и $Y \subseteq R$.

Определение. Пусть $d=\{r_1, r_2, \ldots, r_n\}$ — база данных со схемой $\boldsymbol{R} = \{R_1, R_2, \ldots, R_n\}$ над $\boldsymbol{U}$, $R_i = (S_i, \boldsymbol{K}_i)$, $i=1, \ldots, n$. Она удовлетворяет F, если каждая $X \to Y$ из F^+, применимая к схеме $R_i \in \boldsymbol{R}$, выполняется в отношении $r_i(S_i)$.

Определение. Пусть G — множество всех ФЗ в F^+, которые применимы к какой-нибудь схеме $R_i \in \boldsymbol{R}$. Любая ФЗ в G^+ называется навязанной $\boldsymbol{R}$, а ФЗ из $(F^+ \setminus G^+)$ — ненавязанной $\boldsymbol{R}$. Множество F навязано схеме БД $\boldsymbol{R}$, если каждая ФЗ в F^+ навязана $\boldsymbol{R}$.

Представление множества ФЗ (3)

Определение. *БД d со схемой **R** подчиняется множеству ФЗ F, если F навязано схеме **R** и d удовлетворяет F^+.*

То же самое в виде краткой таблицы:

ФЗ применима к R	если $X \subseteq R$ и $Y \subseteq R$
БД удовлетворяет F	если $\forall\ X \to Y \in F^+$, применимая к R_i, выполняется в r_i
ФЗ $\in G^+$ навязана $\boldsymbol{R}$	если $G \subseteq F^+$, G — все применимые к R_i *ФЗ*
ФЗ не навязана $\boldsymbol{R}$	если *ФЗ* $\in (F^+ \setminus G^+)$
F навязано $\boldsymbol{R}$	если$\forall$ *ФЗ* $\in F^+$ навязана
$d(\boldsymbol{R})$ подчиняется F	если (F навязано $\boldsymbol{R}$) & (d удовлетворяет F^+)

Представление множества ФЗ (4)

Пример. Рассмотрим схему БД $\boldsymbol{R} = \{R_1, R_2, R_3\}$, где $R_1 = ABC$, $R_2 = BCD$, $R_3 = DE$, и множество ФЗ $F = \{A \to BC, C \to A, A \to D, D \to E, A \to E\}$.

Функциональные зависимости $A \to D$ и $A \to E$ не применимы ни к одной схеме из $\boldsymbol{R}$.

Однако множество F навязано схеме $\boldsymbol{R}$, так как существует $G = \{A \to BC, C \to A, C \to D, D \to E\}$, эквивалентное F, каждая функциональная зависимость которого применима к некоторой схеме из $\boldsymbol{R}$. А множество $\{A \to D\}$, не навязано $\boldsymbol{R}$.

Покрытия ФЗ (1)

При работе с базами данных необходимо поддерживать ее корректность: следить, чтобы выполнялись все ФЗ, которым она удовлетворяет.

Поэтому эффективность работы с базой данных зависит от представления функциональных зависимостей. Следовательно, нужно найти такое множество ФЗ, которое, будучи эквивалентным заданному, обладает лучшими в каком-то смысле свойствами.

Рассмотрим методы представления ФЗ, которые позволят упростить эту задачу.

Покрытия ФЗ (2)

Пример. Пусть $F = \{A \to B, B \to C, A \to C, AB \to$ C, $A \to BC\}$, $G = \{A \to B, B \to C\}$. Здесь все ФЗ, принадлежащие F, выводятся из G, то есть $F \equiv G$. Однако представление G предпочтительнее: временная сложность алгоритма оценки множества ФЗ зависит от его объема.

Определение. Множество ФЗ F – покрытие G, если $F \equiv G$.

Определение симметрично относительно множеств ФЗ, каждое из них будет покрытием другого, но обычно подразумевают, что объем покрытия меньше. Так как $F^+ = G^+$, для $\forall\ X \to Y \in G$ выполняется $F \vDash X \to Y$.

Эквивалентность ФЗ

Будем считать, что $F \vDash G$, если $F \vDash X \to Y$ для $\forall\ X \to Y \in G$.

Лемма. Для заданных множеств ФЗ F и G над схемой R эквивалентность $F \equiv G$ имеет место тогда и только тогда, когда $F \vDash G$ и $G \vDash F$.

Доказательство. Пусть $F \equiv G$, тогда для каждой ФЗ $X \to Y$ из F верно $G \vDash X \to Y$, то есть $G \vDash F$, как и для $F \vDash G$.

Пусть теперь $F \vDash G$, тогда $G \subseteq F^+$. Применяя операцию замыкания к обеим частям, получим $G^+ \subseteq (F^+)^+ = F^+$. Аналогично, $(G \vDash F) \Rightarrow (F^+ \subseteq G^+)$. Таким образом, $F^+ = G^+$.

Из леммы следует способ проверки эквивалентности множеств ФЗ на основе алгоритма определения принадлежности ФЗ замыканию множества ФЗ.

Неизбыточные покрытия (1)

Определение. Множество ФЗ F неизбыточно, если у него нет собственного подмножества $F' \subset F$, такого, что $F' \equiv F$. Если F — покрытие G, то F — неизбыточное покрытие G.

Пример. Пусть $G = \{AB \to C, A \to B, B \to C, A \to C\}$, $F = \{AB \to C, A \to B, B \to C\}$. Здесь $F \equiv G$, но F — избыточное покрытие, так как существует $F' = \{A \to B, B \to C\}$, $F' \equiv F$.

ФЗ $X \to Y \in F$ избыточна в F, если $\{F \setminus \{X \to Y\}\} \vDash X \to Y$.

Неизбыточные покрытия (2)

Алгоритм построения неизбыточного покрытия.

Вход: множество функциональных зависимостей *G*.

Выход: его неизбыточное покрытие *F*.

1. Положить F = G.
2. Для очередной ФЗ $X \to Y \in G$ проверить ее принадлежность $\{F \setminus \{X \to Y\}\}^+$.
3. Если $\{F \setminus \{X \to Y\}\} \models X \to Y$, F положить равным $F \setminus \{X \to Y\}$.
4. Повторить с п. 2 до исчерпания множества *G*

Неизбыточные покрытия (3)

Пример. Пусть $F = \{A \to B, B \to A, B \to C, A \to C\}$. Результатом применения алгоритма при выборе функциональных зависимостей в порядке записи будет множество

$$\{A \to B, B \to A, A \to C\}.$$

Если ФЗ выбираются в порядке $A \to B$, $A \to C$, $B \to A$, $B \to C$, то в результате применения алгоритма получим

$$\{A \to B, B \to A, B \to C\}.$$

Посторонние атрибуты (1)

Пусть F — неизбыточное множество ФЗ. В этом случае удаление любой ФЗ приводит к множеству, не эквивалентному F. Значит, за счет уменьшения мощности множества ФЗ уменьшить время обработки не удастся. Но время зависит и от количества атрибутов, входящих в ФЗ.

Определение. *Атрибут $A \in R$ посторонний в $X \to Y \in F$, если выполнено хотя бы одно из следующих условий*:

1. $X = AZ$, $X \neq Z$, $\{F \setminus \{X \to Y\}\} \cup \{Z \to Y\} \equiv F$;
2. $Y = AW$, $Y \neq W$, $\{F \setminus \{X \to Y\}\} \cup \{X \to W\} \equiv F$.

Посторонние атрибуты (2)

Пример. Пусть $F = \{A \to BC, B \to C, AB \to D\}$.

Атрибут C посторонний в правой части $A \to BC$: применяя аксиому проективности (4), получаем $A \to B$.

Атрибут B посторонний в левой части $AB \to D$: положив $X = A$, $Y = B$, $Z = A$ и применив к ($A \to B$, $AB \to D$) аксиому псевдотранзитивности ($X \to Y$, $YZ \to W$) $\Rightarrow$ ($XZ \to W$), получаем $A \to D$.

Посторонние атрибуты (3)

Определение. Функциональная зависимость $X \rightarrow Y \in F$ называется полной или редуцированной слева, если X не содержит атрибута А, постороннего в $X \rightarrow Y$.

Определение. Функциональная зависимость $X \rightarrow Y \in F$ называется редуцированной справа, если Y не содержит атрибута A, постороннего в $X \rightarrow Y$.

Определение. Функциональная зависимость $X \rightarrow Y$ называется редуцированной, если она полная, редуцированная справа и $Y \neq \varnothing$. Множество F называется редуцированным слева (справа), если редуцирована слева (справа) каждая его ФЗ.

Канонические покрытия (1)

Определение. *Неизбыточное множество функциональных зависимостей F, редуцированное слева, называется каноническим, если каждый его элемент имеет вид $X \to A$.*

Заметим, что каждая ФЗ канонического множества редуцирована и справа тоже, поскольку имеет один атрибут справа и F неизбыточно.

Если правый атрибут удалить, образуется ФЗ вида $X \to \varnothing$, которая не влияет на F^+, и ее можно удалить из F, что противоречит неизбыточности. Таким образом, всякое каноническое множество редуцировано.

Канонические покрытия (2)

Определение. Каноническое множество F, которое является покрытием множества G, называется каноническим покрытием множества G.

Пример. $F = \{A \to B, A \to C, A \to D, A \to E, BI \to J\}$ — каноническое покрытие для $G = \{A \to BCE, AB \to DE, BI \to J\}$.

Лемма. Пусть F — редуцированное множество. Образуем G расщеплением каждой ФЗ $X \to A_1A_2 \ldots A_m \in F$ *на* $X \to A_1$, $X \to A_2, \ldots, X \to A_m$. Тогда G — каноническое покрытие F. Обратно, если G — каноническое покрытие F, которое образуется из G объединением функциональных зависимостей с совпадающими левыми частями, то F — редуцированное множество.

Оптимальные покрытия (1)

Определение. Множество ФЗ F минимально, если оно содержит не больше ФЗ, чем любое эквивалентное множество ФЗ.

Минимальное множество F не может содержать избыточных ФЗ, т. е. оно не избыточно.

Пример. Множество $G = \{A \to BC, B \to A, AD \to E, BD \to J\}$ не избыточно, но и не минимально, так как существует $F = \{A \to BC, B \to A, AD \to EJ\}$, $F \equiv G$, которое содержит меньше элементов. В данном случае F минимально.

Определение минимального множества не конструктивно: не задан алгоритм его непосредственного вычисления. Но он существует.

Оптимальные покрытия (2)

Определение. Множество F оптимально, если не существует эквивалентного множества с меньшим числом атрибутных символов.

Пример. Множество $F = \{ EC \to D, AB \to E, E \to AB \}$ — оптимальное покрытие для $G = \{ABC \to D, AB \to E, AB \to E, E \to AB \}$, которое редуцировано и неизбыточно, но не оптимально.

Лемма. Если F — оптимальное множество функциональных зависимостей, то оно редуцировано и минимально.

.

Вторая нормальная форма

Определение. Множество атрибутов *Y* для *ФЗ* $X \to Y \in F^+$ называется полностью зависимым от *X*, если функциональная зависимость $X \to Y$ полная (редуцированная слева). В противном случае множество *Y* называется частично зависимым от *X*.

Определение. Схема отношения R находится во второй НФ относительно множества ФЗ *F,* если для каждого атрибута *A* значения *adom*(*A*) атомарны и каждый непервичный в *R* атрибут полностью зависит от каждого ключа для *R.*

Схема БД ***R*** имеет 2НФ относительно *F*, если каждая схема отношения из ***R*** находится во второй НФ относительно *F*.

Третья нормальная форма

Определение. Схема отношения *R* находится в третьей нормальной форме относительно множества ФЗ *F*, если для каждого атрибута *A* значения *adom*(*A*) атомарны и ни один непервичный атрибут не зависит транзитивно от ключа для *R*.

Утверждение. Любая схема отношения, находящаяся в 3НФ относительно некоторого множества ФЗ, находится во второй нормальной форме относительно того же множества.

Схема БД ***R*** находится в третьей НФ относительно *F*, если каждая схема отношения из ***R*** находится в третьей НФ относительно *F*.

Нормализация через декомпозицию (1)

Схему отношения R, не находящуюся в 3НФ относительно множества ФЗ F, всегда можно разложить в схему БД, находящуюся в третьей НФ относительно F.

Пусть в $R = (S, \boldsymbol{K})$ существует транзитивная зависимость от ключа. Значит, есть ключ $K \in \boldsymbol{K}$, множество $Y \subseteq R$ и непервичный элемент $A \in R$, $A \notin KY$, такие, что $\{K \to Y, Y \to A\} \subseteq F$ и $Y \to K \notin F$.

Определим отношения со схемами R_1 и R_2: $R_1 = R \backslash A$, $R_2 = YA$. Пусть $\boldsymbol{K}$ — множество выделенных ключей для R_1, а $\{Y\}$ — для R_2. Если $\exists K \in \boldsymbol{K}$ такой, что $A \in K$, определим для R_1 новое множество выделенных ключей $\boldsymbol{K'} = \boldsymbol{K} \setminus \{K\} \cup \{K'\}$, где $K' = K \setminus A$.

Нормализация через декомпозицию (2)

Доказано, что для любого $r(R)$, удовлетворяющего F, верно равенство $r = \pi_{R1}(r) \parallel \pi_{R2}(r)$. Таким образом, путем *декомпозиции* R на R_1 и R_2 транзитивная зависимость удалена.

Если в R_1 или в R_2 остались транзитивные зависимости, декомпозиция продолжается. Процесс конечен, так как на каждом шаге получаются все более короткие схемы. Процесс можно ускорить, проверяя $(R \setminus KY)$ на наличие других непервичных атрибутов, зависящих от Y. Такие атрибуты зависят и от K, их можно удалить вместе с A. Пусть это атрибуты $A_1, A_2, \ldots, A_m$. Тогда $R_1 = R \setminus A_1A_2\ldots A_m$, $R_2 = YA_1A_2\ldots A_m$, и отношение R по-прежнему разлагается без потерь на R_1 и R_2.

Нормализация через декомпозицию (3)

Получающаяся в результате декомпозиции схема неоднозначна, она зависит от порядка выбора атрибутов для декомпозиции.

Недостатки алгоритма нормализации через декомпозицию:

1. Большая временная сложность, которая может не ограничиваться полиномиальной.
2. Число порожденных процессом схем отношений может быть больше, чем необходимо.
3. Могут возникать частичные зависимости от ключа, которые, в свою очередь, при декомпозиции порождают больше схем, чем необходимо.
4. Для построенной схемы БД множество ФЗ *F* может оказаться ненавязанным.
5. С помощью этого алгоритма можно получать схемы со «скрытыми» транзитивными зависимостями.

Литература

1. *Атре Ш*. Структурный подход к организации баз данных.
2. *Дейт К*. Руководство по реляционной СУБД DB2.
3. *Каратыгин С. А., Тихонов А. Ф., Тихонова Л. Н.* Visual FoxPro 7.
4. *Кренке Д.* Теория и практика построения баз данных.
5. *Кузнецов С. Д.* Основы баз данных: учебное пособие.
6. *Марков А. С., Лисовский К. Ю*. Базы данных. Введение в теорию и методологию.
7. *Мейер Д.* Теория реляционных баз данных.