Лекция 9

Исчисление доменов.

Функциональные зависимости.

Исчисление доменов

Пусть R – это n-арное отношение с атрибутами A_1, A_2, ..., A_n, тогда в общем виде **<u>условие принадлежности</u>** можно записать так

$R\ (a_{i1} : V_{i1},\ a_{i2} : V_{i2},\ \ldots,\ a_{in} : V_{im})$

где R— имя переменной-отношения, где A— атрибут переменной-отношения R, a_{ij} -значения атрибутов A_j, а V— имя переменной домена или литерально задаваемая константа ($m \le n$).

Условие принадлежности принимает значение **true** в том и только в том случае, если в отношении R существует кортеж, содержащий указанные значения указанных атрибутов a_{ij}.

Если V_{ij} – константа, то на атрибут a_{ij} накладывается жесткое условие, не зависящее от текущих значений доменных переменных; если же V_{ij} – имя доменной переменной, то *условие принадлежности* может принимать разные значения при разных значениях этой переменной.

Пример 6.1. *ППФ исчисления доменов*
СТУДЕНТ(НОМЕР_Л_Д : 1235 , ФИО : ‘Гришин Г. Г.’ , ГРУППА:’У-11’, Специальность: ‘Сист. Управ.’)

База данных Поставщики-Детали

S Поставщики (а)

S#	SNAME	STATUS	CITY
S_1	Smith	20	London
S_2	Jones	10	Paris
S_4	Clark	20	London
S_5	Adams	30	Athens

SP Поставки (в)

S#	P#	QTY
S_1	P_1	300
S_1	P_3	400
S_1	P_6	100
S_2	P_1	300
S_3	P_2	200
S_4	P_4	300
S_4	P_5	400

P Детали (б)

P#	PNAME	COLOR	WEIGHT	CITY
P_1	Nut	Red	12.0	London
P_2	Bolt	Green	17.0	Paris
P_3	Screw	Blue	17.0	Rome
P_4	Screw	Red	14.0	London
P_5	Cam	Blue	12.0	Paris
P_6	Cog	Red	19.0	London

Таблица 6.1 (а, б, в)

Домен	*Переменная домена*
TS#	SX, SY
TP#	PX, PX
TNAME	NAMEX, NAMEY
TCOLOR	COLORX, COLORY
TWEIGHT	WEIGHTX, WEIGHTY
TQTY	QTYX, QTYY
CHAR	CITYX, CITYY
INTEGER	STATUSX, STATUSY

Домены и переменные домена базы данных «Поставщики-Детали» таблица 6.2

Пример 6.2. **Выбрать номера поставщиков из Парижа со статусом, большим 20**

SX WHERE EXISTS STATUSX
(STATUSX > 20 AND S (S#:SX, STATUS:STATUSX, CITY: 'Paris'))

Пример 6.3. **Найти все такие пары номеров поставщиков, в которых два поставщика находятся в одном городе**

(*SX* AS *SA*, *SY* AS *SB*) WHERE EXSIST CITYZ
(S (S#:SX, CITY:CITYZ) AND
S (S#:SY, CITY:CITYZ) AND
SX < SY)

Пример 6.4. **Определить имена поставщиков по крайней мере одной красной детали**

```
NAMEX WHERE EXISTS SX EXISTS PX
          ( S(S#:SX, SNAME:NAMEX) AND
           SP(S#:SX, P#:PX) AND
             P(P#:PX, COLOR:Red) )
```

Пример 6.5. **Определить имена поставщиков всех типов деталей, поставляемых поставщиком с номером 'S_2'**

NAMEX WHERE EXISTS SX EXISTS PX
(S(S#:SX, SNAME:NAMEX) AND
SP(S#:SX, P#:PX) AND
SP(S#:S_2, P#:PX))

Пример 6.6. **Выбрать имена поставщиков всех типов деталей**

```
NAMEX WHERE EXISTS SX (S (S#:SX, SNAME:NAMEX)
              AND FORALL P#   (IF P (P#:PX)
                                   THEN SP (S#:SX, P#:PX)
                              END IF)
```

Пример 6.7. **Определить имена поставщиков, которые не поставляют деталь с номером ‘P_2‘**

NAMEX WHERE EXISTS SX (S (S#:SX, SNAME:NAMEX)
AND NOT SP (S#:SX, P#:P_2))

Пример 6.8. **Получить номера деталей, которые либо весят более 16 фунтов, либо поставляются поставщиком с номером 'S_2', либо и то, и другое**

PX WHERE EXISTS WEIGHTX
(P (P#:PX, WEIGHT:WEIGHTX) AND WEIGHTX > 16.0)
OR SP (S#:S_2, P#:PX)

Функциональные зависимости

Функциональная зависимость (functional dependence или functional dependency) является *связью* между множествами атрибутов внутри данной переменной-отношения.

Выявление ФЗ необходимо для проектирования БД методом **НФ**.

Пусть ***R*** есть переменная-отношение, ***A***— множество атрибутов ***R***, ***X*** и ***Y*** — произвольные подмножества множества А, $X = \{X_1, \ldots, X_k\}$, $Y=\{Y_1, \ldots, Y_m\}$; $p_i(X)$ — кортеж из проекции ***R*** на множество атрибутов ***X***, $p_i(Y)$ — кортеж из проекции ***R*** на множество атрибутов ***Y***; $p_i(X)$, $p_i(Y)$ состоят из элементов кортежа r_i.

Y функционально зависимо от ***X*** ($X \rightarrow Y$) тогда и только тогда, когда для **любого допустимого значения переменной-отношения *R*** и любых двух кортежей $\mathbf{r_1}$ и $\mathbf{r_2}$ из **R** из совпадения $p_1(X)$ *и* $p_2(X)$ следует совпадение $p_1(Y)$ *и* $p_2(Y)$.

Если два кортежа переменной-отношения ***R*** совпадают по значению **X** ($p_1(X) = p_2(X)$), они также совпадают и по значению ***Y*** ($p_1(Y) = p_2(Y)$).

$$X \rightarrow Y \Leftrightarrow (p_1(X) = p_2(X)) \Rightarrow (p_1(Y) = p_2(Y)).$$

Пример 6.16.

SP	***Поставки***	
S#	*P#*	*QTY*
S_1	P_1	300
S_1	P_3	400
S_1	P_6	100
S_2	P_1	300
S_3	P_2	200
S_4	P_4	300
S_4	P_5	400

Таблица 6.3

Для переменной-отношения поставок SP существует ФЗ между множествами атрибутов {S#, P#} и {QTY}

$\{S\#, P\#\} \rightarrow \{QTY\}$.

Пример 6.17.

Переменная-отношение SCP, получена из SP добавлением атрибута CITY

SCP

S#	CITY	P#	QTY
S_1	London	P_1	100
S_1	London	P_2	100
S_2	Paris	P_1	200
S_2	Paris	P_2	200
S_3	Paris	P_2	300
S_4	London	P_2	400
S_4	London	P_4	400
S_4	London	P_5	400

Таблица 6.4 Пример значения переменной-отношения SCP

ФЗ

1. {S#} → {CITY}
2. {S#, P#} → {QTY}
3. {S#, P#} → {CITY}
4. {S#, P#} → {CITY, QTY}
5. {S#, P#} → {S#}
6. {S#, P#} → {S#, P#, CITY, QTY}

7. {S#} → {QTY}
8. {QTY} → {S#}

ФЗ, которые должны быть верны для *любого допустимого значения* переменной-отношения R, могут рассматриваться как **ограничения целостности** переменной-отношения R

Формулировка этого ограничения.

CONSTRAINT S#_CITY_FD

COUNT (SCP {S# }) = COUNT (SCP {S#, CITY})

В любой момент <u>одному поставщику</u> соответствует <u>один город</u>.

Выражение S# $\to$ CITY — сокращенный способ представления этой формулировки

Если X является потенциальным ключом переменной-отношения R, то все атрибуты Y переменной-отношения R должны обязательно быть функционально зависимы от X.

Если переменная-отношение R удовлетворяет ФЗ A $\to$ B и A *не* является потенциальным ключом, то R будет характеризоваться **избыточностью.**

Тривиальные и нетривиальные зависимости

Зависимость A $\rightarrow$ B называется **тривиальной**, если B $\subseteq$ A (множество атрибутов A включает множество B или совпадает с ним).

$$\{S\#, P\#\} \rightarrow S\#$$

Замыкание множества зависимостей

Множество всех ФЗ, логически выводимых из множества ФЗ ***S***, называется **замыканием** множества ФЗ ***S*** и обозначается $\boldsymbol{S^+}$.

Из зависимости {S#, P#}$\rightarrow${CITY, QTY} можно логически вывести следующие ФЗ.

$$\{S\#, P\#\} \rightarrow CITY$$
$$\{S\#, P\#\} \rightarrow QTY$$

Транзитивные зависимости

В отношении P рассмотрим ФЗ

$\{P\#\} \rightarrow \{PNAME\}$ и $\{PNAME\} \rightarrow \{WEIGHT\}$.

Из них можно вывести ФЗ

$\{P\#\} \rightarrow \{WEIGHT\}$

Функциональная зависимость $A \rightarrow C$ называется ***транзитивной***, если существует такой атрибут B, что имеются *функциональные зависимости* $A \rightarrow B$ и $B \rightarrow C$ и отсутствует *функциональная зависимость* $C \rightarrow A$

Пусть A, B и C — произвольные подмножества множества атрибутов заданной переменной-отношения R.

Обозначим объединение множеств A и B как AB $[AB \equiv A \cup B]$.

Правила вывода Армстронга

1. Правило **рефлексивности:** если множество B является подмножеством множества A, то B функционально зависимо от A (если $B \subseteq A$, то $A \rightarrow B$).
2. Правило **пополнения:** если $A \rightarrow B$, то $AC \rightarrow BC$.
3. Правило **транзитивности**: если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$.

Доказательство

1. при $B \subseteq A$ ФЗ $A \rightarrow B$ является *тривиальной.*

2.($A \to B \Rightarrow AC \to BC$)

От противного.

Пусть ФЗ $AC \to BC$ не соблюдается. Это означает, что в некотором допустимом теле отношения найдутся два кортежа t_1 и t_2, такие, что

$$t_1 (AC) = t_2 (AC) \qquad (1),$$

но

$$t_1 (BC) \neq t_2 (BC) \qquad (2).$$

Здесь t (A) — проекция кортежа t на множество атрибутов A.

$A \subseteq AC$ $[AC \equiv A \cup C]$, $\Rightarrow$ по *аксиоме* рефлексивности $\underline{AC \to A}$.

Следовательно, по определению ФЗ: $t_1(AC)=t_2(AC) \Rightarrow t_1(A)=t_2(A)$.

Т.к. $\exists$ ФЗ $A \to B$, то $t_1(A)=t_2(A) \Rightarrow t_1(B)=t_2(B)$.

Тогда из $t_1(B)=t_2(B)$ & $t_1(BC) \neq t_2(BC)$ *(2)* $\Rightarrow$ $t_1(C) \neq t_2(C)$, что противоречит наличию *тривиальной* ФЗ $AC \to C$ ($t1(AC)=t2(AC) \Rightarrow t1(C)=t2(C)$).

Следовательно, предположение об отсутствии ФЗ $AC \to BC$ неверно.

Второе правило вывода доказано

3. $(A \to B$ и $B \to C \Rightarrow A \to C)$

Пусть существуют ФЗ $A \to B$ и $B \to C$, но $A \to C$ не имеет места. Последнее означает, что в некотором допустимом теле отношения найдутся два кортежа t_1 и t_2, такие, что $t_1(A) = t_2(A)$, но $t_1(C) \neq t_2(C)$.

Из наличия ФЗ $A \to B$ $\Rightarrow$ из $t_1(A)=t_2(A)$ $\Rightarrow$ $t_1(B)=t_2(B)$.

Из наличия ФЗ $B \to C$ $\Rightarrow$ из $t_1(B)=t_2(B)$ $\Rightarrow$ $t_1(C)=t_2(C)$,

что противоречит предположению об отсутствии ФЗ $A \to C$.

$$(\exists\ A \to B \wedge \exists\ B \to C) \Rightarrow$$

$$[(\exists\ t_1, t_2) \mid (t_1(A)=t_2(A) \Rightarrow t_1(B)=t_2(B))\ \&\ (t_1(B)=t_2(B) \Rightarrow t_1(C)=t_2(C))] \Rightarrow$$

$$\Rightarrow [t_1(A)=t_2(A) \Rightarrow t_1(C)=t_2(C)]$$

Третье правило *вывода* доказано

Система *правил вывода* Армстронга **полна** и **совершенна**

Дополнительные правила.

4. Правило **самодетерминированности**: $A \to A$.
5. Правило **декомпозиции**: если $A \to BC$, то $A \to B$ и $A \to C$.
6. Правило **объединения**: если $A \to B$ и $A \to C$, то $A \to BC$.
7. Правило **композиции**: если $A \to B$ и $C \to D$, то $AC \to BD$.

где D —некоторое произвольное подмножество множества атрибутов R

Доказательство

Правило 4 По правилу 1 ($B \subseteq A \Rightarrow A \to B$): ($A \subseteq A \Rightarrow A \to A$).

Правило 5. По правилу 1 ($B \subseteq A \Rightarrow A \to B$) : $B \subseteq BC \Rightarrow BC \to B$.

По правилу 3 ($A \to B$ & $B \to C \Rightarrow A \to C$): ($\mathbf{A \to BC}$ & $BC \to B$) $\Rightarrow \mathbf{A \to B}$.

$C \subseteq BC \Rightarrow BC \to C$, ($\mathbf{A \to BC}$ & $BC \to C$) $\Rightarrow \mathbf{A \to C}$.

Правило 6. По правилу 2 ($A \to B \Rightarrow AC \to BC$): $\mathbf{A \to B} \Rightarrow AA \to AB$,

что равносильно $A \to AB$.

$\mathbf{A \to C} \Rightarrow AB \to CB$, т.к. $BC = CB$, то $AB \to BC$.

По правилу 3: ($A \to AB$ & $AB \to BC$) $\Rightarrow \mathbf{A \to BC}$.

Правило 7. $\mathbf{A \to B} \Rightarrow AC \to BC$: $\mathbf{C \to D} \Rightarrow BC \to BD$.

$AC \to BC$ & $BC \to BD$ по правилу 3 получаем $AC \to BD$

Правила вывода

1. Правило **рефлексивности:** $B \subseteq A \Rightarrow A \to B$
2. Правило **пополнения:** $A \to B \Rightarrow AC \to BC$
3. Правило **транзитивности**: $A \to B$ и $B \to C \Rightarrow A \to C$
4. Правило **самодетерминированности**: $A \to A$
5. Правило **декомпозиции**: $A \to BC \Rightarrow A \to B$ и $A \to C$
6. Правило **объединения**: $A \to B$ и $A \to C \Rightarrow A \to BC$
7. Правило **композиции**: $A \to B$ и $C \to D \Rightarrow AC \to BD$

Пример 6.18.

Пусть дана некоторая переменная-отношение ***R*** с **атрибутами *A***, ***B***, ***C***, ***D***, ***E***, ***F*** и имеют место следующие функциональные зависимости.

S=$\{\{A\}\to\{B, C\}, \{B\}\to\{E\}, \{C, D\}\to\{E, F\}\}$

Покажем, что для переменной-отношения ***R*** выполняется функциональная зависимость $\{A, D\}\to\{F\}$, которая принадлежит к замыканию заданного множества функциональных зависимостей.

1. Дана ФЗ $\{A\}\to\{B,C\}$

2. Из ФЗ $\{A\}\to\{B,C\}$ по (A→BC ⇒ A→B и A→C (5)) ⇒ ФЗ $\boldsymbol{\{A\}\to\{C\}}$;

3. Из ФЗ $\{A\}\to\{C\}$ по (A→B ⇒ AC→BC (2)) ⇒ ФЗ $\boldsymbol{\{A,D\}\to\{C,D\}}$;

4. ФЗ $\boldsymbol{\{C,D\}\to\{E, F\}}$ —дана по условию;

5. Из ФЗ $\{A, D\}\to\{C,D\}$ и ФЗ $\{C,D\}\to\{E,F\}$ по (A→B и B→C ⇒ A→C (3)) ⇒ ФЗ $\boldsymbol{\{A,D\}\to\{E, F\}}$;

6. Из ФЗ $\{A,D\}\to\{E, F\}$ по (A→BC ⇒ A→B и A→C (5)) ⇒ $\{A,D\}\to\{F\}$;

Алгоритм вычисления замыкания Z^+.

Дано: 1. Отношение ***R***.

2. Множество ***Z*** атрибутов отношения ***R***

(подмножество заголовка ***R***, или составной атрибут ***R***), **|Z|=m**.

3. Множество функциональных зависимостей ***S*** вида ***X*** $\to$ ***Y***, **|S|=n**

Найти: замыкания Z^+

Алгоритм.

CloseZ := **Z**; // на0–м шаге замыкание равно самому множеству Z

Repeat

CloseZ_Old := CloseZ; // сохраняем старое значение замыкания

For J:=1 To n Do

Begin

Взять очередную *j-ю* ФЗ ***X***$\to$***Y*** из ***S***;

If **X** $\subseteq$ CloseZ // проверяем *является ли* X *подмножеством* CloseZ

Then CloseZ := CloseZ UNION **Y**; // Да, добавляем в множество

// CloseZ новый элемент Y

End;

Until CloseZ =CloseZ_Old; // на очередном шаге ничего не добавили

Z⁺:= CloseZ;

Пример 6.19.

Дано

1. Отношение R с заголовком {A, B, C, D, E F}
2. Множество атрибутов Z={A,B}
3. Множество ФЗ S ={ {A}→{B,C}, {E}→{C,F}, {B}→{E},{C,D}→{E,F}}

Требуется найти $\{A,B\}^{+}$ над S.

1. Начальное значение CloseZ = {A,B}.
2. Выполним внутренний цикл *For* четыре раза —для каждой ФЗ из S.

- 1) {A}→{B,C}, **{A}** ⊂ **{A,B}** ⇒ C добавляется в CloseZ, {B}⊂{A,B};

  CloseZ = {A,B,C};
- 2) {E}→{C,F}, **{E}** ⊄ **{A,B,C}** ⇒ CloseZ не изменяется;
- 3) {B}→{E}, **{B}** ⊂ **{A,B,C}** ⇒ E добавляется в CloseZ;

  CloseZ = {A,B,C,E};
- 4) {C,D}→{E,F}, **{C,D}**⊄**{A,B,C,E}** ⇒ CloseZ не изменяется;

  Внутренний цикл завершен.

4. Значение замыкания CloseZ = {A,B,C,E}, CloseZ $\neq$ CloseZ_Old.

Внешний цикл Repeat повторяется еще раз.

Внутренний цикл выполняется четыре раза.

1) $\{A\}\to\{B,C\}$, $\{A\} \subset \{A,B,C,E\}$, но $\{B,C\}\subset\{A,B,C,E\} \Rightarrow$
CloseZ не изменяется;

2) $\{E\}\to\{C,F\}$, $\{E\} \subset \{A,B,C,E\}$, $\Rightarrow$ F добавляется в CloseZ;
CloseZ ={**A,B,C,E,F**};

3) $\{B\}\to\{E\}$, $\{B\} \subset \{A,B,C,E, F\}$, но $\{E\} \subset\{A,B,C,E,F\} \Rightarrow$
CloseZ не изменяется ;

4) $\{C,D\}\to\{E,F\}$, $\mathbf{\{C,D\}\not\subset\{A,B,C,E,F\}}$ $\Rightarrow$ CloseZ не изменяется;

5. CloseZ ={A,B,C,E,F}, CloseZ $\neq$ CloseZ_Old.

Внешний цикл Repeat повторяется, но в процессе работы цикла замыкание CloseZ останется неизменным.

6. Алгоритм завершит работу с результатом $\mathbf{\{A,B\}^+=\{A,B,C,E,F\}}$.

Пример 6.20

Дано:

1. Отношение R с заголовком {A, B, C, D, E, F};
2. Множество атрибутов отношения ***R*** Z **=** {A,E}
3. Множество ФЗ

   $S = \{\{A\}\to\{D\}, \{A,B\}\to\{E\}, \{B,F\}\to\{E\}, \{C,D\}\to\{F\}, \{E\}\to\{C\}\}$.

Требуется найти $\{A,E\}^+$ над S.

1. Начальное значение CloseZ= {A,E}.
2. Внутренний цикл For выполняется пять раз — для каждой ФЗ из S.

- 1) $\{A\}\to\{D\}$, $\mathbf{\{A\}} \subset \mathbf{\{A,E\}}$ $\Rightarrow$ D добавляется в CloseZ;

  CloseZ = {A,D,E};
- 2) $\{A,B\}\to\{E\}$, $\mathbf{\{A,B\}} \not\subset \mathbf{\{A,D,E\}}$ $\Rightarrow$ CloseZ не изменяется;
- 3) $\{B,F\}\to\{E\}$, $\mathbf{\{B,F\}} \not\subset \mathbf{\{A,D,E\}}$ $\Rightarrow$ CloseZ не изменяется;
- 4) $\{C,D\}\to\{F\}$, $\mathbf{\{C,D\}} \not\subset \mathbf{\{A,D,E\}}$ $\Rightarrow$ CloseZ не изменяется;
- 5) $\{E\}\to\{C\}$), $\mathbf{\{E\}} \subset \mathbf{\{A,D,E\}}$ $\Rightarrow$ C добавляется в CloseZ;

  CloseZ= {A,C,D,E};

<u>Внутренний цикл завершен</u>.

3. Значение замыкания CloseZ = {A,C,D,E} $\Rightarrow$ CloseZ $\neq$ CloseZ_Old.

   Внешний цикл Repeat повторяется еще раз.

   Внутренний цикл For выполняется пять раз.

   На первых трех шагах результат останется прежним.

- 1) $\{A\}\rightarrow\{D\}$, $\{A\}\subset\{A,C,D,E\}$, но $D \subset \{A,C,D,E\}$;
- 2) $\{A,B\}\rightarrow\{E\}$, $\{A,B\}\not\subset\{A,C,D,E\}$;
- 3) $\{B,F\}\rightarrow\{E\}$, $\{B,F\}\not\subset\{A,C,D,E\}$;
- 4) $\{C,D\}\rightarrow\{F\}$, $\{C,D\}\subset\{A,C,D,E\}$ $\Rightarrow$ F добавляется в CloseZ;

  CloseZ ={A,C,D,E,F};
- 5) $\{E\}\rightarrow\{C\}$, $\{E\}\subset\{A,C,D,E,F\}$, но $C \subset \{A,C,D,E,F\}$;

4. CloseZ={A,C,D,E,F} $\Rightarrow$ CloseZ $\neq$ CloseZ_Old.

   Внешний цикл Repeat повторяется. Внутренний цикл выполняется пять раз, замыкание CloseZ останется неизменным.

5. Процесс завершится с результатом **$\{A,E\}^+=\{A,C,D,E,F\}$**.

Суперключ ***K*** переменной-отношения ***R*** — это подмножество атрибутов ***R***, которое содержит, по крайней мере, один *потенциальный* ключ.

ФЗ $K \rightarrow A$ будет истинна для каждого атрибута ***R***.

Множество ***K*** является суперключом тогда и только тогда, когда замыкание K^+ для множества ***K*** в пределах заданного множества ФЗ является множеством абсолютно <u>всех</u> атрибутов переменной-отношения ***R***.

Неприводимые множества зависимостей

Пусть S_1 и S_2 — два множества ФЗ.

Если любая ФЗ, которая выводится из S_1, выводится из S_2 ($S_1^+ \subseteq S_2^+$), то множество S_2 называется **покрытием** для множества S_1.

Если S2 является покрытием для S_1, а S_1 является покрытием для S_2 ($S_1^+ = S_2^+$), то множества S_1 и S_2 **эквивалентны.**

Множество функциональных зависимостей S называется **неприводимым** тогда и только тогда, когда оно обладает следующими свойствами.

1. Правая (зависимая) часть каждой ФЗ из множества S содержит только один атрибут (т.е. является *одноэлементным* множеством).
2. Левая часть (детерминант) каждой ФЗ из множества S является *неприводимой*, т.е. ни один *атрибут* из детерминанта не может быть опущен без изменения замыкания S^+ (без конвертирования множес-тва S в некоторое иное множество, не эквивалентное множеству S). В этом случае ФЗ называется **неприводимой слева.**
3. Ни одна ФЗ из множества S не может быть удалена из множества S без изменения его замыкания S^+ (т.е. без конвертирования множества S в иное множество, не эквивалентное множеству S).

Множество ***K*** является <u>потенциальным ключом</u> тогда и только тогда, когда оно является <u>неприводимым суперключом</u>

Пример 6.21.

Рассмотрим переменную-отношение деталей P со следующими ФЗ.

$P\# \rightarrow PNAME$

$P\# \rightarrow COLOR$

$P\# \rightarrow WEIGHT$

$P\# \rightarrow CITY$

Это множество ФЗ является <u>неприводимым</u>.

Правая часть каждой зависимости содержит только один атрибут, а левая часть является неприводимой.

Ни одна из перечисленных ФЗ не может быть опущена без изменения замыкания множества (т.е. без *утраты некоторой информации).*

Примеры множеств ФЗ не являющихся неприводимыми.

1. P#→{PNAME, COLOR}
   P#→WEIGHT
   P#→CITY

(Правая часть первой ФЗ *не является одноэлементным множеством*.)

2. (P#, PNAME)→COLOR
   P#→PNAME
   P#→WEIGHT
   P#→CITY

(Первую ФЗ можно упростить, опустив атрибут PNAME в левой части без изменения замыкания, т.е. она *не является неприводимой слева*.)

3. *P#→P#*
   P#→PNAME
   P#→COLOR
   P#→WEIGHT
   P#→CITY.

Утверждение.

Для любого множества ФЗ существует по крайней мере одно эквивалентное множество, которое является неприводимым $\mathbf{S^-}$.

Пусть дано исходное множество зависимостей S.

Можно без утраты общности предположить, что каждая ФЗ в множестве S имеет одноэлементную правую часть
(по правилу декомпозиции $(\{A\}\to\{B,C\} \Rightarrow \{A\}\to\{B\}$ *и* $\{A\}\to\{C\})$).

Для каждой зависимости **f** из множества **S** следует проверить каждый атрибут $\mathbf{A}_i$ в левой части зависимости **f**.
Если множество **S** и множество зависимостей, полученное в результате устранения атрибута $\mathbf{A_i}$ в левой части зависимости **f** эквивалентны, этот атрибут следует удалить.

Для каждой оставшейся в множестве S зависимости f, если множества S и S\f эквивалентны, удалить зависимость f из множества S.
Получившееся в результате таких действий множество S^- является неприводимым и эквивалентно исходному множеству S.

Множество ФЗ ***I***, которое неприводимо и эквивалентно другому множеству ФЗ S, называется **неприводимым покрытием множества S**.

Пример 6.22.

Пусть дана переменная-отношение R с атрибутами A, B, C, D и следующими ФЗ.

1. $\{A\}\to\{B,C\}$
2. $\{B\}\to\{C\}$
3. $\{A\}\to\{B\}$
4. $\{A,B\}\to\{C\}$
5. $\{A,C\}\to\{D\}$

Найдем неприводимое множество ФЗ, эквивалентное данному множеству.

1. Перепишем заданные ФЗ таким образом, чтобы каждая из них имела одноэлементную правую часть.

   $\{A\}\to\{B\}$

   $\{A\}\to\{C\}$

   $\{B\}\to\{C\}$

   $\{A\}\to\{B\}$

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

   $\{A,C\}\to\{D\}$

Зависимость $\{A\}\to\{B\}$ записана дважды, $\Rightarrow$ одну можно удалить.

2. Атрибут **C** левой части зависимости $\{\mathbf{A,C}\}\rightarrow\{D\}$ может быть опущен:

1) $\exists$ $\mathbf{\{A\} \rightarrow\{C\}}$, из $\{AA\} \rightarrow\{AC\} \Rightarrow\{A\}\rightarrow\{A,C\}$ (по правилу пополнения) .

2) $\exists$ $\mathbf{\{A,C\}\rightarrow\{D\}}$, из $\{A\}\rightarrow\{A,C\}$ и $\{A,C\}\rightarrow\{D\} \Rightarrow\{A\}\rightarrow\{D\}$ (по правилу транзитивности).

3) Получаем $\mathbf{A\rightarrow D}$

3. Зависимость $\{A,B\}\rightarrow\{C\}$ может быть исключена:

1)$\exists$ $\mathbf{\{A\}\rightarrow\{C\}}$, из $\{A\}\rightarrow\{C\} \Rightarrow\{A,B\}\rightarrow\{C,B\}$ (по правилу пополнения).

2) Из $\{A,B\}\rightarrow\{C,B\} \Rightarrow \{A,B\}\rightarrow\{C\}$ (по правилу декомпозиции) .

4. Зависимость$\{A\}\rightarrow\{C\}$ может быть исключена:

Из $\{A\}\rightarrow\{B\}$ и $\{B\}\rightarrow\{C\} \Rightarrow \{A\}\rightarrow\{C\}$ (по правилу транзитивности).

5. В результате получили неприводимое множество зависимостей.

$$\mathbf{\{A\rightarrow B,\ B\rightarrow C,\ A\rightarrow D\}}$$