

Содержание

1. Операции реляционной алгебры	2
2. Определение функциональной зависимости в теории проектирования реляционных баз данных. Аксиомы Армстронга.	4
3. Правила объединения, декомпозиции и псевдотранзитивности в теории проектирования реляционных баз данных (доказательство леммы).	5
4. Алгоритм вычисления замыкания множества атрибутов в теории проектирования реляционных баз данных.	6
5. Алгоритм построения минимального покрытия для множества функциональных зависимостей.	7
6. Свойство соединения без потерь для схемы базы данных.	8
7. Свойство сохранения зависимостей для схемы базы данных.	10
8. Ключ отношения. Алгоритм построения ключа.	11
9. Третья нормальная форма.	12
10. Алгоритм построения «хорошей» схемы базы данных. Преимущества и недостатки алгоритма.	14
11. Пример построения «хорошей» схемы БД.	16
12. Пример построения «хорошей» схемы БД - 2.	19

1. Операции реляционной алгебры

Схема отношения – поименованное множество атрибутов.

$R = (A_1, \dots, A_n)$, где A_i ($i=1, n$) - атрибут

Пример: $R = (\text{имя, адрес, товар, цена}) = (A_1, A_2, A_3, A_4)$

Степень схемы отношения – количество атрибутов в схеме.

Отношение (экземпляр отношений) – конкретная таблица с заданной схемой отношения.

Пример

Имя	адрес	Товар	Цена
АО «х»	г. Москва, ул. у, д. z	Сахар	25
АО «х1»	г. Москва, ул. у1, д. z 1	карамель	30

← Схема отношения

Кортеж – строка отношения.

Определение схемы БД

Пусть A – множество всех атрибутов предметной области (универсальная схема отношения).

$R_1, \dots, R_n$ – схемы отношения такие, что $\bigcup_{i=1}^n R_i = A$

В этом случае говорят, что $r = (R_1, \dots, R_n)$ – **схема БД**.

1. Объединение 2-х отношений

$r_3 = r_1 \vee r_2$ – отношение, каждый кортеж которого принадлежит либо отношению r_1 , либо r_2 .

r1

=

A1	A2
1	2
3	4

r2

=

A1	A2
5	6
1	2

r3

=

A1	A2
1	2
3	4
5	6

В реляционной алгебре дублирование кортежей не допускается.

r_1 и r_2 должны обладать одинаковыми схемами отношения.

2. Разность отношений

$r_3 = r_1 - r_2$ – отношение, каждый кортеж которого принадлежит отношению r_1 , но не принадлежит r_2 .

r_1, r_2 – см. выше

r_3	<table><tr><th>A1</th><th>A2</th></tr><tr><td>3</td><td>4</td></tr></table>	A1	A2	3	4
A1	A2				
3	4				

3. Пересечение отношений

$r_3 = r_1 \cap r_2$ – отношение, каждый экземпляр которого принадлежит и отношению r_1 , и отношению r_2 .

r_1, r_2 – см. выше

r_3	<table><tr><th>A1</th><th>A2</th></tr><tr><td>1</td><td>2</td></tr></table>	A1	A2	1	2
A1	A2				
1	2				

4. Декартово произведение

Пусть R и S – две схемы отношения со степенями k1 и k2 (степень – число атрибутов в схеме отношения). r, s – соответствующие экземпляры отношений.

Тогда $t = r \times s$ – декартово произведение, т.е. отношение со степенью $k1 + k2$, каждый кортеж которого получается путем конкатенации кортежей из r и s.

Порядок кортежей в отношениях r и s неважно.

A1	A2	r		s		A1	A3
1	2					5	6
3	4					7	8

R.A1	R.A2	S.A1	S.A3	t
1	2	5	6	
1	2	7	8	
3	4	5	6	
3	4	7	8	

5. Проекция отношений на некоторое подмножество атрибутов

$t = \pi_{A_1..A_k}(r)$ – отношение, каждый кортеж которого состоит из значений атрибутов $A_1..A_k$ кортежа из r.

A1	A2	A3	A4	r
1	2	3	4	
7	8	9	10	
3	4	5	6	
3	4	7	6	

A1	A2	$\pi_{A1,A2}(r)$
1	4	
7	10	
3	6	
3	6	

6. Селекция

$t = \sigma_F(r)$ – отношение, каждый кортеж которого принадлежит r и удовлетворяет условию F.

A1	A2	r
1	2	
9	8	
3	3	

A1	A2	$t = \sigma_{A1 \leq A2}(r)$
1	2	
3	3	
3	3	

7. Естественное соединение

$t = r \bowtie s$, r и s – экземпляры отношений со схемами R и S.

Определение следует из правил его построения:

- 1) построить декартово произведение $r \times s$;
- 2) из декартова произведения выбрать кортежи, удовлетворяющие условию:

$$R.A_1 = S.A_1 \dots R.A_k = S.A_k$$

$A_{i1}, \dots, A_{ik}$ - атрибуты, общие в схемах отношений R и S.

3) удалить дублирующие столбцы $S_0.A_{i1}, \dots, S_1.A_{ik}$ и получим требуемое отношение t.

r	A ₁	A ₂	A ₃	s	A ₁	A ₂	A ₄
	1	2	3		1	2	7
	4	6	7		8	9	10

Построить соединение $t = r \bowtie s$

1) $r \times s$

R.A ₁	R.A ₂	R.A ₃	S.A ₁	S.A ₂	S.A ₄
1	2	3	1	2	7
1	2	3	8	9	10
4	6	7	1	2	7
4	6	7	8	9	10

1	2	3	1	2	7
---	---	---	---	---	---

3) t:	A ₁	A ₂	A ₃	A ₄
	1	2	3	7

2. Определение функциональной зависимости в теории проектирования реляционных баз данных. Аксиомы Армстронга.

1) Функциональная зависимость

Пусть R определение функциональной зависимости. $R = (A_1, \dots, A_n)$ - универсальная схема отношений, т.е. схема состоит из всех атрибутов предметной области.

X и Y – некое подмножество этой схемы;

$X \rightarrow Y$ (X функционально определяет Y), если в любом экземпляре r не существует двух кортежей, совпадающих по атрибутам X и не совпадающих по атрибутам Y.

Пример Задана универсальная схема отношений:

$R = (\text{имя, адрес, товар, цена}) = (A_1, A_2, A_3, A_4)$

и функциональные зависимости:

$A_1 \rightarrow A_2, A_1 A_3 \rightarrow A_4$

Пусть в БД хранится единственный экземпляр данной схемы:

r =	A ₁	A ₂	A ₃	A ₄
	АО «Х»	г. Москва, ул. y, д. z	Сахар	25
	АО «Х»	г. Москва, ул. y, д. z	Карамель	30
	АО «Х»	г. Москва, ул. y, д. z	Пастила	40

$A_1 A_3 \rightarrow A_4$ имеет место, так как нет кортежей, которые совпадали бы по этой паре атрибутов (по X и Y).

2) Замыкание множества функциональных зависимостей

Пусть R – универсальная схема отношения;

F – заданное множество функциональных зависимостей на этой схеме.

Замыканием F назовем множество функциональных зависимостей, которые логически следуют из F .

Функциональная зависимость логически следует из F , если её можно получить с помощью аксиом Армстронга. Обозначается F^+ .

Аксиомы Армстронга (правила вывода)

- 1) Аксиомы рефлексивности: если $Y \subseteq X \subseteq R$, то $X \rightarrow Y$
- 2) Аксиома пополнения: пусть $X \rightarrow Y$, $Z \subseteq R$ (Z может быть $\emptyset$), тогда $X \cup Z \rightarrow Y \cup Z$ ($XZ \rightarrow YZ$)
- 3) Аксиома транзитивности: если $X \rightarrow Y$, $Y \rightarrow Z$, то $X \rightarrow Z$

Пример построения замыкания.

(A,B,C)

$F=(A \rightarrow B, B \rightarrow C)$

F^+ -?

- 1) Построение тривиальных функциональных зависимостей:
 $A \rightarrow A, B \rightarrow B, C \rightarrow C, AB \rightarrow A, AB \rightarrow B, AC \rightarrow A, AC \rightarrow C, BC \rightarrow C, BC \rightarrow B, ABC \rightarrow A, ABC \rightarrow C, ABC \rightarrow B, ABC \rightarrow AC, ABC \rightarrow AB, ABC \rightarrow BC, ABC \rightarrow ABC, AB \rightarrow AB, AC \rightarrow AC, BC \rightarrow BC$
- 2) Используем вторую аксиому Армстронга
 $A \rightarrow AB, AC \rightarrow BC, B \rightarrow BC$
 $AB \rightarrow AC, AC \rightarrow ABC, AB \rightarrow ABC$
- 3) Используем третью аксиому Армстронга
Т.к. $A \rightarrow B, B \rightarrow C$, то $A \rightarrow C$
Аналогично, $AB \rightarrow BC, A \rightarrow AC, A \rightarrow ABC$

Теорема

Аксиомы Армстронга являются надежными и полными.

Надежность: если функциональная зависимость выводится с помощью аксиомы Армстронга, то она справедлива в любых экземплярах отношений, где справедливы исходные функциональные зависимости.

Полнота: если имеет место функциональная зависимость $X \rightarrow Y$, то она обязательно может быть выведена с помощью аксиом Армстронга.

3. Правила объединения, декомпозиции и псевдотранзитивности в теории проектирования реляционных баз данных (доказательство леммы).

Справедливы следующие правила реляционной алгебры:

- 1) Правило объединения
Если множество атрибутов X определяет множество атрибутов Y ($X \rightarrow Y$) и $X \rightarrow Z$, то $X \rightarrow YZ$
- 2) Правило декомпозиции:
Если $X \rightarrow Y, Z \subseteq Y$, то $X \rightarrow Z$
- 3) Правило псевдотранзитивности:
Если $X \rightarrow Y, WY \rightarrow Z$, то $WX \rightarrow Z$

Доказательство:

1. $X \rightarrow XY$ (пополняем атрибутом X первую зависимость)
 $XY \rightarrow YZ$ (пополняем атрибутом Y вторую зависимость)
По третьей аксиоме Армстронга получаем $X \rightarrow YZ$
2. Т.к. $X \rightarrow Y$, $Z \subseteq Y$, то по первой аксиоме Армстронга $Y \rightarrow Z$. Тогда по 3 аксиоме получаем: $X \rightarrow Z$
3. $WX \rightarrow WY$ (пополняем атрибутом W первую зависимость)
 $WY \rightarrow Z$ (по условию). Тогда по 3 аксиоме получаем: $WX \rightarrow Z$

4. Алгоритм вычисления замыкания множества атрибутов в теории проектирования реляционных баз данных.

Замыкание множества функциональных зависимостей

F – заданное множество функциональных зависимостей на этой схеме.

Количество функциональных зависимостей, входящих в замыкание F^+ , может быть очень велико, даже если число исходных функциональных зависимостей незначительно.

Пример

$$F = \{X \rightarrow A_1, \dots, X \rightarrow A_n\}$$

Из правила объединения следует, что справедлива функциональная зависимость $X \rightarrow Y$, где Y- произвольное подмножество множества $(A_1, \dots, A_n)$. Таких комбинаций может быть 2^n .

Поэтому в теории нормализации схем отношений напрямую замыкание F^+ не строится, но есть подход, позволяющий определить, принадлежит ли функциональная зависимость $X \rightarrow Y$ замыканию F^+ . Этот подход основывается на понятии замыкания множества атрибутов.

Замыкание множества атрибутов.

Пусть R- универсальная схема отношений. Замыканием множества атрибутов X, принадлежащего R ($X \subseteq R$), называется множество атрибутов $A_{i1}, \dots, A_{ik}$, таких, что справедливо следующее выражение:

$$(X \rightarrow A_{i1}, \dots, X \rightarrow A_{ik}) \subseteq F^+.$$

Замыкание множества атрибутов обозначается через X^+ .

Правило

Чтобы проверить, принадлежит ли функциональная зависимость $X \rightarrow Y$ замыканию F^+ ($X \rightarrow Y \subseteq F^+$), сначала строят X^+ и если $Y \subseteq X^+$, то значит $X \rightarrow Y \subseteq F^+$.

Основная задача заключается в построении алгоритма замыкания множества атрибутов.

Алгоритм построения замыкания множества атрибутов

Это итерационная процедура, включающая следующие шаги:

1. $i:=0$, $X_0^+ = X$ – замыкание множества атрибутов на i-ом шаге.
2. Положить $X_{i+1}^+ = X_i^+ \cup V$, где V – множеств атрибутов такое, что в F имеется функциональная зависимость $Y \rightarrow Z$, для которых $Y \subseteq X_i^+$, а $V \subseteq Z$.
3. $X_{i+1}^+ = X_i^+$ – это есть $X^+ := X_i^+$, иначе $i:=i+1$ и перейти к шагу 2.

Пример замыкания множества атрибутов.

$$R = (A, B, C, D, E, G),$$

$$F = (AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C, CG \rightarrow BD, CE \rightarrow AG)$$

$X=BD$, X^+ - ?

1. $X_0^+=BD$
2. Оформим в виде таблицы:

I	$Y \rightarrow Z$, для которых $Y \subseteq X_i^+$	$V \subseteq Z$	$X_{i+1}^+ := X_i^+ \cup V$
0	$D \rightarrow EG$	EG	EGBD X_1^+
1	$BE \rightarrow C$	C	EGBDC X_2^+
2	$C \rightarrow A$, $BC \rightarrow D$, $CG \rightarrow BD$, $CE \rightarrow AG$	A	EGBDCA X_3^+
3	$AB \rightarrow C$, $ACD \rightarrow B$		EGBDCA X_4^+

Т.о. $(BD)^+ = EGBDCA^+ \Rightarrow$

$(BD \rightarrow E, BD \rightarrow G, BD \rightarrow B, BD \rightarrow C, BD \rightarrow D, BD \rightarrow A) \subset F^{+**}$

* и ** эквивалентны.

5. Алгоритм построения минимального покрытия для множества функциональных зависимостей.

Определение покрытия. Пусть F и G - множества функциональных зависимостей. G называют покрытием F, если их замыкания совпадают. $G^+ = F^+$

Определение минимального покрытия. Покрытие G называется минимальным, если оно содержит:

- а) минимальное число функциональных зависимостей
- б) минимальное число атрибутов в левой и правых частях функциональных зависимостей.

Алгоритм построения минимального покрытия G для множества функциональных зависимостей F:

- 1) Каждую функциональную зависимость из F заменить на совокупность функциональных зависимостей, каждая из которых содержит 1 атрибут в правой части. Полученное множество функциональных зависимостей обозначим через G.
- 2) Для каждой функциональной зависимости $(X \rightarrow A) \in G$ выполнить следующие действия:
 - а. Проверить, принадлежит ли данная функциональная зависимость $(G-X \rightarrow A)^+$ и если да, то исключить её из G (т.к. её можно вывести из меньшего числа функциональных зависимостей).

Примечание. Чтобы выполнить эту проверку, необходимо построить замыкание множества атрибутов на множестве функциональных зависимостей $(G-X \rightarrow A)$ и если $A \in (G-X \rightarrow A)^+$ и проверить, принадлежит ли A...

- б. Для каждого собственного подмножества $Z < X$ проверить $Z \rightarrow A \in G^+$, и если да, то заменить зависимость $X \rightarrow A$ на $Z \rightarrow A$ (с меньшим числом атрибутов), повторить п.2б, где вместо $X \rightarrow A$ выступает $Z \rightarrow A$.

Примечание. Чтобы проверить, принадлежит ли $Z \rightarrow A \in G^+$, необходимо построить Z^+ для множества функциональных зависимостей G и проверить $A \in G^+$, и если да, что по определению $Z \rightarrow A \in G^+$

- 3) Зависимости в G с одинаковой левой частью объединить в 1 функциональную зависимость (см. правило объединения доказанных выше лемм).

п.п.1,2 алгоритма обеспечивают минимальное число атрибутов в левой и правой частях каждой функциональной зависимости из G. п.3 обеспечивает минимальное число функциональных зависимостей G.

Пример построения минимального покрытия.

Пусть задана универсальная схема отношений $R=(A,B,C,D)$

$F=AB \rightarrow CD, BC \rightarrow A, BC \rightarrow D, C \rightarrow D$.

Построить минимальное покрытие G от F.

- 1) Заменить каждую функциональную зависимость (правило декомпозиции) $G=(AB \rightarrow D, AB \rightarrow C, BC \rightarrow A, BC \rightarrow D, C \rightarrow D)$.

$AB \rightarrow D, G-AB \rightarrow D=(AB \rightarrow C, BC \rightarrow A, BC \rightarrow D, C \rightarrow D)$

$(AB)^+ = ABCD$ (D принадлежит замыканию $\Rightarrow AB \rightarrow D \in (G-AB \rightarrow D)^+ \Rightarrow$ эту зависимость можно исключить, она может быть выведена из меньшего числа).

$G=(AB \rightarrow C, BC \rightarrow A, BC \rightarrow D, C \rightarrow D)$.

- 2) $AB \rightarrow C, G-AB \rightarrow C=(BC \rightarrow A, BC \rightarrow D, C \rightarrow D)$. Можем ли вывести эту зависимость из меньшего числа.

$(AB)^+ = AB, AB \rightarrow C \notin (G-AB \rightarrow C)^+$

- 3) $BC \rightarrow A, G-BC \rightarrow A=(AB \rightarrow C, BC \rightarrow D, C \rightarrow D)$

$(BC)^+ = BCD - A$ не вошло $\Rightarrow BC \rightarrow A$ нельзя вывести $\Rightarrow BC \rightarrow A \notin (G-BC \rightarrow A)^+$

- 4) $BC \rightarrow D, G-BC \rightarrow D=(AB \rightarrow C, BC \rightarrow A, C \rightarrow D)$.

$(BC)^+ = BCAD \Rightarrow BC \rightarrow D \in (G-BC \rightarrow D)^+$, т.е. мы эту зависимость можем исключить.
 $G=(AB \rightarrow C, BC \rightarrow A, C \rightarrow D)$.

- 5) Рассмотрим $C \rightarrow D$. Можно ли эту зависимость вывести из меньшего числа.

$C \rightarrow D, G-C \rightarrow D=(AB \rightarrow D, BC \rightarrow A)$. $C^+ = C \Rightarrow C \rightarrow D \notin (G-C \rightarrow D)^+$.

- 1) $G=(AB \rightarrow C, BC \rightarrow A, C \rightarrow D)$ (*)

$AB \rightarrow C$

a. $A \rightarrow C, A^+ = A \Rightarrow A \rightarrow C \notin G^+$

b. $B \rightarrow C, B^+ = B \Rightarrow B \rightarrow C \notin G^+$

- 2) $BC \rightarrow A$. Таким же образом можно убедиться, что

$B \rightarrow A \notin G^+$

$C \rightarrow A \notin G^+$

Таким образом, * - минимальное покрытие и рассмотренный алгоритм – основа построения хорошей схемы БД.

6. Свойство соединения без потерь для схемы базы данных.

Определение свойства соединения без потерь: Пусть $\rho = (R_1, \dots, R_n)$ – схема БД (т.е.

$\bigcup_{i=1}^n R_i = A$, где A – универсальная схема отношения). Схема БД обладает **свойством**

соединения без потерь, если для любого экземпляра отношения r универсальной схемы отношений A выполняется следующее равенство:

$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \dots \bowtie \pi_{R_n}(r)$, где $\pi_{R_i}(r)$ – проекция отношения r на множество атрибутов R_i , $\bowtie$ – операция естественного соединения.

Пример схемы БД, не обладающей свойством соединения без потерь.

Пусть $R = (A, B, C)$ – универсальная схема отношения, $\rho = (AB, BC) = (R_1, R_2)$ – схема БД, а $F = (A \rightarrow B)$ – функциональная зависимость.

Для доказательства утверждения, что схема БД не обладает свойством соединения без потерь, достаточно привести пример экземпляра отношения r , который не удовлетворял бы определению свойства соединения без потерь.

$$\begin{array}{c}
 \begin{array}{ccc} A & B & C \\ r = a_1 & b_1 & c_1, \\ & a_2 & b_1 & c_2 \\ A & B & C \\ a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_1 & c_2 \end{array}
 \end{array}
 \quad \text{тогда} \quad
 \begin{array}{c}
 \begin{array}{cc} A & B \\ \pi_{R_1}(r) = a_1 & b_1; \\ & a_2 & b_1 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{cc} B & C \\ \pi_{R_2}(r) = b_1 & c_1; \\ & b_1 & c_2 \end{array}
 \end{array}$$

$$\pi_{R_1}(r) \bowtie \pi_{R_2}(r) = a_1 \quad b_1 \quad c_2 \neq r.$$

Пример схемы БД, обладающей свойством соединения без потерь.

Пусть $R = (A, B, C)$ – универсальная схема отношения, $\rho = (AB, AC) = (R_1, R_2)$ – схема БД, а $F = (A \rightarrow B)$ – функциональная зависимость.

Покажем, в пику предыдущему примеру, что для следующего экземпляра отношения выполняется свойство соединения без потерь:

$$\begin{array}{c}
 \begin{array}{ccc} A & B & C \\ r = a_1 & b_1 & c_1; \\ & a_2 & b_1 & c_2 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{cc} A & B \\ \pi_{R_1}(r) = a_1 & b_1; \\ & a_2 & b_1 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{cc} A & C \\ \pi_{R_2}(r) = a_1 & c_1; \\ & a_2 & c_2 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{ccc} A & B & C \\ \pi_{R_1}(r) \bowtie \pi_{R_2}(r) = a_1 & b_1 & c_1 = r. \\ & a_2 & b_1 & c_2 \end{array}
 \end{array}$$

Теорема: пусть $\rho = (R_1, R_2)$ – схема БД, а F – множество функциональных зависимостей на универсальной схеме отношения.

ρ обладает свойством соединения без потерь тогда и только тогда, когда справедлива хотя бы одна из следующих функциональных зависимостей:

- 1) $R_1 \cap R_2 \rightarrow R_1 - R_2$;
- 2) $R_1 \cap R_2 \rightarrow R_2 - R_1$.

Покажем, что в предыдущем примере схема БД ρ обладает свойством соединения без потерь.

$$R_1 \cap R_2 = A;$$

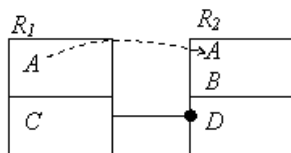
$$R_1 - R_2 = B;$$

$$R_1 \cap R_2 \rightarrow R_1 - R_2 \ (A \rightarrow B), \text{ т.к. } A \rightarrow B \text{ из } F \text{ (по условию).}$$

Следствие из теоремы: пусть R_1 и R_2 – две сущности инфологической схемы БД, которые связаны между собой.

Схема БД $\rho = (R_1, R_2)$ обладает свойством соединения без потерь, если общие атрибуты R_1 и R_2 содержат ключ одной из этих сущностей.

Пример:



Обладает ли $\rho = (R_1, R_2)$ свойством соединения без потерь?

Воспользуемся теоремой:

$$R_1 \cap R_2 = A, \quad R_1 - R_2 = C$$

$$R_1 \cap R_2 \rightarrow R_1 - R_2, \text{ т.к. имеет место } A \rightarrow C \text{ (т.к. } A \text{ — ключ } R_1).$$

Если связь между сущностями неидентифицирующая, все было бы то же самое.

7. Свойство сохранения зависимостей для схемы базы данных.

Определение проекции множества функциональных зависимостей F на схему отношения R_i .

Пусть $\rho = (R_1, \dots, R_n)$ – схема БД, F – исходное множество функциональных зависимостей.

Проекцией F на R_i называется множество функциональных зависимостей $X \rightarrow Y \in F^+$ таких, что имеет место следующее включение: $XY \subseteq R_i$.

Проекция F на R_i обозначается: $\Pi_{R_i}(F)$.

Определение свойства сохранения зависимости

Схема БД ρ обладает **свойством сохранения зависимости**, если справедливо следующее равенство:

$$\left(\bigcup_{i=1}^n \Pi_{R_i}(F) \right)^+ = F^+.$$

Данное выражение читается следующим образом:

- 1) Берется $\Pi_{R_i}(F)$.
- 2) Полученные функциональные зависимости объединяются.
- 3) Для этих зависимостей строится замыкание (применяются аксиомы Армстронга).
- 4) Полученное множество функциональных зависимостей сравнивается с F^+ .

Пример схемы БД, не обладающей свойством сохранения зависимости.

Пусть $R = (A, B, C)$ – универсальная схема отношения, $\rho = (AB, BC) = (R_1, R_2)$ – схема БД, а $F = (AB \rightarrow C, C \rightarrow A)$ – функциональная зависимость.

Доказать, что ρ не обладает свойством сохранения зависимости.

$$\Pi_{R_1=AB}(F) = (A \rightarrow A, B \rightarrow B, AB \rightarrow A, AB \rightarrow B, AB \rightarrow AB, B \rightarrow A, B \rightarrow AB);$$

$$\Pi_{R_2=BC}(F) = (B \rightarrow B, C \rightarrow C, BC \rightarrow B, BC \rightarrow C, BC \rightarrow BC).$$

Найдем такую функциональную зависимость, которая принадлежит F^+ , но не принадлежит левой части выражения в определении свойства сохранения зависимости. И тем самым докажем, что ρ не обладает свойством соединения без потерь.

В качестве такой зависимости имеем:

$$AB \rightarrow C \in F^+ \text{ (по определению).} \quad (**)$$

$$AB \rightarrow C \notin \left(\bigcup_{i=1}^2 \Pi_{R_i}(F) \right)^+. \quad (*)$$

Для доказательства последнего утверждения требуется построить $(AB)^+$ на объединенном множестве функциональных зависимостей и проверить $C \in (AB)^+$?

$(AB)^+ = AB$, но $C \notin AB$.

Это доказывает утверждение (*).

(*) и (**) доказывают утверждение задачи (что ρ не обладает свойством сохранения зависимости), т.е. $\left(\bigcup_{i=1}^2 \Pi_{R_i}(F) \right)^+ \neq F$.

Пример схемы БД, обладающей свойством сохранения зависимости.

Пусть $R = (A, B, C)$ – универсальная схема отношения, $\rho = (R)$ – схема БД, а $F = (AB \rightarrow C, C \rightarrow A)$ – функциональная зависимость.

Доказать, что ρ обладает свойством сохранения зависимости.

$\Pi_R(F)$ – все зависимости, полученные из F с помощью аксиом Армстронга.

Т.о. равенство доказано и данная схема БД обладает свойством сохранения зависимости.

8. Ключ отношения. Алгоритм построения ключа.

Пусть $R = (A_1, \dots, A_n)$ – некоторая схема отношения, F – множество функциональных зависимостей на этой схеме.

Тогда множество атрибутов $X \subseteq R$ называется **ключом** схемы отношения R , если выполняются следующие условия:

- 1) $X \rightarrow A_1 \dots A_n \in F^+$.
- 2) $\forall Y \subset X, Y \rightarrow A_1 \dots A_n \notin F^+$.

Т.е. ключ содержит в себе минимальное число атрибутов, обладающих свойством 1.

Из определения следует, что ключ в таблице всегда уникален (иначе это привело бы к тому, что таблица содержала бы крайней мере две одинаковых записи).

Алгоритм построения ключа

1. $i := 0$, $X_0 := A_1, \dots, A_n$ – ключ на 0 шаге.
2. ДЛЯ по числу атрибутов в X_i , $j = \overline{1, n}$
 ЕСЛИ $(X_i - A_j)^+ = R$, то $X_{i+1} := X_i - A_j$, $i++$, $n--$, выйти из цикла
 КОНЕЦ ДЛЯ
3. ЕСЛИ i – возросло, то перейти к п. 2 алгоритма
 ИНАЧЕ $X := X_j$ – ключ схемы отношения $R = (A_1, \dots, A_n)$.

Примечание: Алгоритм позволяет построить один ключ. В общем случае ключей в схеме отношения может быть несколько.

Определение 1. Если какой либо атрибут A входит в состав какого-либо ключа схемы отношения, то он называется первичным.

Определение 2. Если атрибут A не входит ни в какой ключ схемы отношения, то он называется не первичным.

Пример построения ключа

Дано: $R = ABCD$ – универсальная схема отношения; $F = (AB \rightarrow CD, BC \rightarrow AD)$ – функциональные зависимости.

Задача: X – ключ R ?

Решение:

1. $i := 0, X_0 := ABCD$
2. $(BCD)^+ = BCDA, X_1 := BCD, i = 1, n = 3$
3. i – возросло
2. $(CD)^+ = CD \neq R \quad (BD)^+ = BD \neq R$
 $(BC)^+ = BCDA = R, X_2 := BC, i = 2, n = 2$
3. i – возросло
2. $(C)^+ = C \neq R \quad (B)^+ = B \neq R$
3. i – не возросло, $X = BC$ – ключ.

Примечание: В данном варианте – это не единственный ключ (AB – тоже ключ). Всё зависит от последовательности просмотра атрибутов.

$X = AB$ – тоже ключ; ABC – первичные атрибуты; D – не первичный атрибут.

9. Третья нормальная форма.

Схема отношения R находится в 3НФ, если не существует ключа X , множества $Y \subseteq R$, и не первичного атрибута $H (H \notin Y)$ для которых выполнялись бы следующие условия:

1. $X \rightarrow Y \in F^+$
2. $Y \rightarrow H \in F^+$
3. $Y \rightarrow X \notin F^+$

Примечание: Из определения следует, что если можно подобрать X, Y и H для которых выполняются условия 1–3, то схема отношения не находится в 3НФ.

Пример доказательства, что схема отношения НЕ находится в 3НФ

Дано: Используя определение 3НФ доказать, что схема отношения не находится в 3НФ.

$R = (\text{имя}, \text{адрес}, \text{товар}, \text{цена}) = (A, B, C, D)$ – универсальная схема отношения;
 $F = (A \rightarrow B, AC \rightarrow D)$ – функциональные зависимости.

Задача: Доказать что R не находится в 3НФ.

Решение:

Определим ключ.

1. $i := 0, X_0 := ABCD$
2. $(BCD)^+ = BCD \neq R$
 $(ACD)^+ = ACDB = R, X_1 := ACD, i = 1, n = 3$
3. i – возросло
2. $(CD)^+ = CD \neq R \quad (AD)^+ = ADB \neq R$
 $(AC)^+ = ACBD = R, X_2 := AC, i = 2, n = 2$
3. i – возросло

$$2. (C)^+ = C \neq R \qquad (A)^+ = AB \neq R$$

3. i – не возросло, $X = AC$ – ключ схемы отношения.

Примечание: Можно показать, что он единственный.

- Выбираем X, Y, H .

$$X = AC \qquad Y = A \qquad H = B \notin Y = A \text{ – не первичный атрибут.}$$

- Проверяем условия 1–3, которые указаны в определении 3НФ.

1. $X \rightarrow Y$, так как $AC \rightarrow A$ (первая аксиома рефлексивности Армстронга);

2. $Y \rightarrow H$, так как $A \rightarrow B$ (по условию задачи);

3. $Y \not\rightarrow X$, так как $A \not\rightarrow AC$ (для доказательства последнего утверждения построим замыкание левой части $(A)^+$: $(A)^+ = AB$, $AC \not\subseteq AB$, поэтому $A \not\rightarrow AC$).

- Таким образом удалось построить тройку X, Y, H , для которой выполнялись бы условия 1–3 из определения 3НФ.

Поэтому R не находится в 3НФ и обладает всеми аномалиями.

Пример доказательства, что схема отношения находится в 3НФ

Дано: Используя определение 3НФ доказать, что схема отношения находится в 3НФ.

$R = (\text{имя, адрес, товар, цена}) = (A, B, C, D)$ – универсальная схема отношения;

$F = (A \rightarrow B, AC \rightarrow D)$ – функциональные зависимости; $\rho = (AB, ACD) = (R_1, R_2)$ – схема БД.

Задача: Доказать что R_1, R_2 находятся в 3НФ.

Решение:

1. $R_1 = AB$

а) $X = A$ – ключ, так как $A \rightarrow B$ по условию задачи;

б) подберём Y для которого $X \rightarrow Y$, $Y \not\rightarrow X$

$$\underline{Y = B} \quad ((B)^+ = B, A \not\subseteq B, B \not\rightarrow A);$$

в) Здесь нельзя подобрать не первичный атрибут H , который $H \notin Y$ (единственный не первичный атрибут – B , но $B \in Y$).

Мы не сумели подобрать X, Y, H для которых были бы справедливы условия 1–3 из определения 3НФ, следовательно, по определению, R_1 находится в 3НФ.

2. $R_2 = ACD$

1) $X = AC$ – ключ

2) найдём Y для которого $X \rightarrow Y$, $Y \not\rightarrow X$

Легко можно убедиться, что таким условиями удовлетворяют следующие Y -ки:

- | | |
|---------|---------|
| а) A | б) C |
| в) D | г) AD |
| д) CD | |

3) Попытаемся подобрать не первичный атрибут H , для которого $H \notin Y$, $Y \rightarrow H$.

Единственным претендентом является D , так как AC – ключ, D – не первичный.

а) $Y = A$, но $Y \rightarrow H$, потому что $A \not\rightarrow D$, $(A)^+ = AB$, $D \not\subseteq AB$ (D не входит в AB)

б) $Y = C$, но $Y \rightarrow H$, потому что $C \not\rightarrow D$, $(C)^+ = C$, $D \not\subseteq C$

в-д) не проходят, так как здесь не первичный атрибут (D) $H \in Y$.

Методом логических исключений мы пришли к выводу, что здесь нельзя подобрать X , Y , H для которых были бы справедливы условия 1–3 из определения 3НФ, следовательно, по определению, R_1 находится в 3НФ. Поэтому вывод – R_2 находится в 3НФ.

10. Алгоритм построения «хорошей» схемы базы данных. Преимущества и недостатки алгоритма.

Схема БД называется «хорошей», если она обладает следующими свойствами:

1. Свойством соединения без потерь.
2. Свойством сохранения зависимости.
3. Свойством нормализации схем отношений схемы БД.

Примечание: последнее свойство обеспечивает отсутствие аномалий: избыточности, потенциальной противоречивости, аномалии включения записи, аномалии удаления

Алгоритм построения «хорошей» схемы БД

Дано: $R = (A_1, \dots, A_n)$ – универсальная схема отношения; F – исходное множество функциональных зависимостей на R .

Задача: Построить «хорошую» схему БД. ρ – ?

Решение:

1. $\rho = \emptyset$
2. Определить минимальное покрытие G для функциональных зависимостей F (см. предыдущие лекции).
3. Каждую функциональную зависимость V вида $V \rightarrow W$ из G заменить на VW (объединить атрибуты левой и правой частей). Получившееся множество схем отношений обозначить Q .
4. Если $R = (A_1, \dots, A_n) \in Q$, то добавить в ρ схему отношения $R = (A_1, \dots, A_n)$ и выйти из алгоритма, (ρ – «хорошая» схема БД).
Иначе перейти к пункту 5.
5. Добавить в ρ в качестве схем отношений все те одиночные атрибуты, которые не вошли ни в одну из схем из Q .
6. Добавить в ρ все схемы отношений из Q .

Примечание: после выполнения пунктов алгоритма 1-6 схема БД обладает Свойством Сохранения Зависимостей, и каждая её схема отношения находится в 3НФ.

7. Если ни одна из схем отношений в ρ не содержит в себе ключ для универсальной схемы отношения R , то добавить в ρ какой-нибудь ключ для R в качестве схемы отношения.

Примечание: после выполнения всех 7 пунктов алгоритма схема БД обладает всеми свойствами «хорошей» схемы БД:

- 1) Свойство Соединения Без Потерь;
- 2) Свойство Сохранения Зависимостей;
- 3) Каждая схема отношения в ρ находится в 3НФ.

7-е условие слишком жесткое (надо найти ключ для универсальной схемы отношения – а в ней бывают сотни атрибутов).

Примечание: 7-ой пункт предполагает, что, в принципе, при разработке таблиц в БД, будут соединены все таблицы из ρ , но на практике такого не бывает, поэтому его можно ослабить:

7. Если в запросах к БД ρ соединяет не все схемы отношений из ρ , а только из подмножества $Q_1 \subseteq \rho$, $Q_2 \subseteq \rho$, ..., $Q_k \subseteq \rho$, то в ρ можно добавить ключи для Q_1 , Q_2 , ..., Q_k в качестве новых схем отношений. (Если, конечно, эти ключи не содержатся уже ни в одной из схем отношения из ρ).

Пример построения «хорошей» схемы БД

Пусть $R = (\text{имя}, \text{адрес}, \text{товар}, \text{цена}) = (A, B, C, D)$; $F = (A \rightarrow B, BC \rightarrow D, AC \rightarrow D, B \rightarrow A)$

Задача: Построить хорошую схему БД – ρ .

Воспользуемся алгоритмом:

1. $\rho = \emptyset$
2. Воспользуемся алгоритмом построения минимального покрытия
 - $G = (A \rightarrow B, BC \rightarrow D, AC \rightarrow D, B \rightarrow A)$
 - 2.2.1.1 $A \rightarrow B$, $G - A \rightarrow B = (BC \rightarrow D, AC \rightarrow D, B \rightarrow A)$
 $(A)^+ = A$, $A \rightarrow B \notin (G - A \rightarrow B)^+$
 - 2.2.1.2 $BC \rightarrow D$, $G - BC \rightarrow D = (A \rightarrow B, AC \rightarrow D, B \rightarrow A)$
 $(BC)^+ = BCAD \Rightarrow BC \rightarrow D \in (G - BC \rightarrow D)^+$ Данную функциональную зависимость исключаем из G
 $G = (A \rightarrow B, AC \rightarrow D, B \rightarrow A)$
 - 2.2.1.3 $AC \rightarrow D$, $G - AC \rightarrow D = (A \rightarrow B, B \rightarrow A)$
 $(AC)^+ = ACB \Rightarrow AC \rightarrow D \notin (G - AC \rightarrow D)^+$
 - 2.2.1.4 $B \rightarrow A$, $G - B \rightarrow A = (A \rightarrow B, AC \rightarrow D)$
 $(B)^+ = B$, $B \rightarrow A \notin (G - B \rightarrow A)^+ \Rightarrow G = (A \rightarrow B, AC \rightarrow D, B \rightarrow A)$
 - 2.2.2 В дальнейшем замыкания множества атрибутов рассматривается для G
 - 2.2.2.1 $AC \rightarrow D$
 - a) $A \rightarrow D$
 $(A)^+ = AB$, $A \rightarrow D \notin (G)^+$ - (Если это верно, то можно заменить $AC \rightarrow D$ на $A \rightarrow D$. Док-во: $AC \rightarrow A$, $A \rightarrow D \Rightarrow AC \rightarrow D$, тогда $AC \rightarrow D$ можно заменить на $A \rightarrow D$)
 - b) $C \rightarrow D$
 $(C)^+ = C$, $C \rightarrow D \notin (G)^+$
 $G = (A \rightarrow B, AC \rightarrow D, B \rightarrow A)$ – минимальное покрытие.
3. $Q = (AB, ACD, BA) = (AB, ACD)$
4. Нужно убедиться, что $R \notin Q$. Так оно и есть: $R = ABCD \notin Q = (AB, ACD)$.
5. ---
6. $\rho = (AB, ACD)$.
7. Определим ключ для R
 - 7.1. $i := 0$; $x_0 = BACD$ (изменение в последовательности символов связано с известным заранее результатом)
 - 7.2. $(ACD)^+ = ACDB = R$, $x_1 = ACD$, $i++$;
 - 7.3. $i := 1$;
 - 7.2. $(CD)^+ = CD \neq R$
 - $(AD)^+ = ADB \neq R$
 - $(AC)^+ = ACBD = R$, $x_2 = AC$, $i++$;

7.3. $i=2$;
 7.2. $(C)^+ = C \neq R$;
 $(A)^+ = A \neq R$;
 7.3. $i=2$;
 $x = x_2 = AC$ – ключ R .
 $AC \in ACD$ в $\rho = (AB, ACD) \Rightarrow$ В ρ ничего не надо добавлять, и поэтому $\rho = (AB, ACD)$ – хорошая схема БД.

Преимущества и недостатки построения хорошей схемы БД.

Преимущества «+»

- Алгоритм определяет стандартную процедуру построения схемы БД, обладает свойствами соединения без потерь, сохранения зависимостей и нормализации отношений.
- Можно использовать для доказательства, что схема БД оптимальна. (может понадобиться на дипломе)

Недостатки «-»

- Очень сложно определить все множество физических зависимостей предметной области (Алгоритм критичен к набору физических зависимостей $\rightarrow$ неустойчив)
- При увеличении числа физических зависимостей сложность вычислений резко возрастает.
- Синтез хорошей схемы БД, как правило, приводит к увеличению схем отношений в схемах БД. Чтобы объединить данные в схемах отношений необходимо в операторах SQL использовать соединение таблиц, а это приводит к увеличению времени выполнения запроса.

Указанные недостатки сдерживают применение рассмотренного метода на практике. Но существует практические приемы нормализации, которые позволяют обойти эти недостатки.

11. Пример построения «хорошей» схемы БД.

Постановка:

Пусть задана универсальная схема отношений R :

$R = (A$ – читаемый курс (дисциплина), B – преподаватель, C – час начала занятий, D – аудитория, E – студент, T – оценка)

И задано множество функциональных зависимостей:

$F = \{$

$CD \rightarrow B$	в аудитории одновременно может быть только один преподаватель;
$AC \rightarrow D$	каждый курс одновременно может читаться только в одной аудитории;
$CE \rightarrow A$	каждый студент одновременно может слушать только один курс;
$A \rightarrow B$	каждый курс ведет только один преподаватель;
$CD \rightarrow A$	в аудитории в один и тот же момент времени может читаться только один курс;

$CB \rightarrow D$	преподаватель в один и тот же момент времени может находиться только в одной аудитории;
$AE \rightarrow T$	по каждому курсу каждый студент имеет только одну оценку;
$CE \rightarrow D$	студент в один и тот же момент времени может находиться только в одной аудитории

}

Построить ρ – «хорошую» схему БД.

Решение:

1. Пусть $\rho = 0$ - множество схем отношений, которые будут находиться в хорошей БД.

2. Найдем минимальное покрытие для $F(G)$:

а)

$$1) G = (CD \rightarrow B, AC \rightarrow D, CE \rightarrow A, A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

2)

$$CD \rightarrow B : G - CD \rightarrow B = (AC \rightarrow D, CE \rightarrow A, A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

$$(CD)^+ = CDAB, B \in (CD)^+ \Rightarrow CD \rightarrow B \in (G - CD \rightarrow B)^+ \Rightarrow$$

$$G = (AC \rightarrow D, CE \rightarrow A, A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

3)

$$AC \rightarrow D : G - AC \rightarrow D = (CE \rightarrow A, A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

$$(AC)^+ = ACBD, D \in (AC)^+ \Rightarrow AC \rightarrow D \in (G - AC \rightarrow D)^+ \Rightarrow$$

$$G = (CE \rightarrow A, A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

4)

$$CE \rightarrow A : G - CE \rightarrow A = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

$$(CE)^+ = CEDATB, A \in (CE)^+ \Rightarrow CE \rightarrow A \in (G - CE \rightarrow A)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

5)

$$A \rightarrow B : G - A \rightarrow B = (CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

$$(A)^+ = A, B \notin (A)^+ \Rightarrow A \rightarrow B \notin (G - A \rightarrow B)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

6)

$$CD \rightarrow A : G - CD \rightarrow A = (A \rightarrow B, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

$$(CD)^+ = CD, A \notin (CD)^+ \Rightarrow CD \rightarrow A \notin (G - CD \rightarrow A)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

7)

$$CB \rightarrow D : G - CB \rightarrow D = (A \rightarrow B, CD \rightarrow A, AE \rightarrow T, CE \rightarrow D)$$

$$(CB)^+ = CB, D \notin (CB)^+ \Rightarrow CB \rightarrow D \notin (G - CB \rightarrow D)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

8)

$$AE \rightarrow T : G - AE \rightarrow T = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, CE \rightarrow D)$$

$$(AE)^+ = AEB, T \notin (AE)^+ \Rightarrow AE \rightarrow T \notin (G - AE \rightarrow T)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

9)

$$CE \rightarrow D : G - CE \rightarrow D = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T)$$

$$(CE)^+ = CE, D \notin (CE)^+ \Rightarrow CE \rightarrow D \notin (G - CE \rightarrow D)^+ \Rightarrow$$

$$G = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

b)

$$CD \rightarrow A, C^+ = C, A \notin C^+, D^+ = D, A \notin D^+$$

$$CB \rightarrow D, C^+ = C, D \notin C^+, B^+ = B, D \notin B^+$$

$$AE \rightarrow T, A^+ = AB, T \notin A^+, E^+ = E, T \notin E^+$$

$$CE \rightarrow D, C^+ = C, D \notin C^+, E^+ = E, D \notin E^+$$

с) Зависимости с одинаковыми левыми частями отсутствуют

$$G_{\min} = (A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow T, CE \rightarrow D)$$

3. $Q = (AB, CDA, CBD, AET, CED)$

4. $R = ABCDET \notin Q$

5. Все атрибуты вошли

6. $\rho = (CED, AB, CDA, CBD, AET)$

7. Определим ключ для R

$$1). i = 0, X_0 = ABCDET$$

$$2). (BCDET)^+ = BCDETA = R, X_1 = BCDET, i = 1$$

$$3). i++$$

$$2). (CDET)^+ = CDETBA = R, X_2 = CDET, i = 2$$

$$3). i++$$

$$2). (DET)^+ = DET \neq R$$

$$(CET)^+ = CETDAB = R, X_3 = CET, i = 3$$

$$3). i++$$

$$2). (ET)^+ = ET \neq R$$

$$(CT)^+ = CT \neq R$$

$$(CE)^+ = CETBDA = R, X_4 = CE, i = 4$$

$$3). i++$$

$$2). E^+ = E \neq R$$

$$C^+ = C \neq R, X_4 = CE, i = 4$$

$X = CE, CE \in CEA \subset Q$ (ключ в полученную схему отношений не надо добавлять)

Полученная схема отношений $\rho = (CED, AB, CDA, CBD, AET) = (R_1, R_2, R_3, R_4, R_5)$ обладает всеми свойствами «хорошей» схемы БД (соединение без потерь - позволяет делать правильную выборку запросами, сохранение зависимостей, находится в третьей нормальной форме - исключает избыточность, потенциальную противоречивость, аномалию включения и удаления записей).

Ответ: $\rho = (CED, AB, CDA, CBD, AET)$

Подставляя исходные данные, получим пять таблиц искомой БД:

- {час начала занятий, аудитория, преподаватель}
- {курс, час начала занятий, аудитория}
- {курс, преподаватель}
- {курс, студент, оценка}
- {час начала занятий, студент, аудитория}

12. Пример построения «хорошей» схемы БД - 2.

Предположим, у нас есть поставщики, которые поставляют некоторые товары.

supplier (id(A), name(B))

goods(id(C), name(D))

contract(goods_id(C), supplier_id(A), price(E))

Поставщик может поставлять разные товары. Один товар может поставляться разными поставщиками. В скобках указаны условные обозначения.

Докажем, что у нас «хорошая» схема БД и найдем ее ключ.

Решение:

1. Пусть $\rho = 0$ - множество схем отношений, которые будут находиться в хорошей БД.

2. Найдем минимальное покрытие для F (G):

а)

$$1) G = (A \rightarrow B, C \rightarrow D, CA \rightarrow E)$$

2)

$$A \rightarrow B : G - A \rightarrow B = (C \rightarrow D, CA \rightarrow E)$$

$$(A)^+ = A, B \notin (A)^+ \Rightarrow A \rightarrow B \notin (G - A \rightarrow B)^+ \Rightarrow$$

$$G = (A \rightarrow B, C \rightarrow D, CA \rightarrow E)$$

3)

$$C \rightarrow D : G - C \rightarrow D = (A \rightarrow B, CA \rightarrow E)$$

$$(C)^+ = C, D \notin (C)^+ \Rightarrow C \rightarrow D \notin (G - C \rightarrow D)^+ \Rightarrow$$

$$G = (A \rightarrow B, C \rightarrow D, CA \rightarrow E)$$

4)

$$CA \rightarrow E : G - CA \rightarrow E = (A \rightarrow B, C \rightarrow D)$$

$$(CA)^+ = CABD, E \notin (CA)^+ \Rightarrow CA \rightarrow E \notin (G - CA \rightarrow E)^+ \Rightarrow$$

$$G = (A \rightarrow B, C \rightarrow D, CA \rightarrow E)$$

b)

$$CA \rightarrow E, C^+ = CD, E \notin C^+, A^+ = AB, E \notin A^+$$

с) Зависимости с одинаковыми левыми частями отсутствуют

$$G_{\min} = (A \rightarrow B, C \rightarrow D, CA \rightarrow E)$$

$$8. Q = (AB, CD, CAE)$$

$$9. R = ABCDE \notin Q$$

10. Все атрибуты вошли

$$11. \rho = (AB, CD, CAE)$$

12. Определим ключ для R

$$1). i = 0, X_0 = ABCDE$$

$$2). (ABCD)^+ = ABCDE = R, X_1 = ABCD, i = 1$$

$$i++$$

$$2). (ABC)^+ = ABCDE = R, X_2 = ABC, i = 2$$

$$i++$$

$$3). (AB)^+ = AB \neq R$$

$$(AC)^+ = ABCDE = R, X_3 = AC, i = 3$$

$$i++$$

$$4). (A)^+ = AB \neq R$$

$$(C)^+ = CD \neq R$$

$X = AC = CA, CA \in CAE \subset Q$ (ключ в полученную схему отношений не надо добавлять)

Ответ: $\rho = (AB, CD, CAE)$, мы доказали, что разработанная нами схема БД является «хорошей». Ключ схемы – AC.