

БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Ст.преп. каф. ИТ Кеберле Наталья Геннадьевна

**Лекция 9. Недостатки реляционной модели данных:
избыточность данных, аномалии удаления и обновления.
Понятие о нормализации схемы базы данных.
Функциональные зависимости между данными.**





На прошлой лекции

- ▶ Мы познакомились со структурой и ограничениями целостности реляционной модели данных
- ▶ Какова эта структура?
- ▶ Что такое ограничение целостности?
- ▶ Какие ограничения целостности есть в реляционной модели?
- ▶ Что такое Null-значение? Есть ли альтернатива Null-значениям?



Недостатки реляционной модели

- ▶ Иногда реляционные отношения содержат **избыточную** информацию, или данные скомпонованы таким образом, что при обновлении данных в одном отношении нарушается целостность всей базы
- ▶ Такие ситуации возникают если при построении базы данных **не** учитывается или учитывается **не полностью** информация о зависимостях между атрибутами внутри отношений



Недостатки реляционной модели

- ▶ Хранение повторяющихся значений данных чаще всего ведет:
 - к избыточности хранения (и к увеличению нужного места хранения)
 - к аномалиями удаления и обновления



Пример избыточности хранения

график (

пилот	рейс	дата	время_вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

Дублирование сочетаний:

Рейс - время_вылета

Пример избыточности хранения

график (

пилот	рейс	дата	время_вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

Дублирование сочетаний:
пилот - рейс

Последствия избыточности

- ▶ Избыточность данных провоцирует появление аномалий **обновления**:

график (

пипот	рейс	дата	время_вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

Обновляя время_вылета рейса 281, придётся, для поддержки целостности, просмотреть всё отношение *график* и исправить время вылета во всех кортежах с атрибутом «рейс», равным 281

Последствия избыточности

- ▶ Избыточность данных провоцирует появление аномалий **удаления**:

график (

пилот	рейс	дата	время_вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

Если рейс 412 выполнял единственный пилот, и этого пилота перевели на другие рейсы, то при удалении кортежа <Коупли, 412, 15.авг., 13:25> информация об этом рейсе теряется

Последствия избыточности

- ▶ Избыточность данных провоцирует появление аномалий **добавления**:

график (

пилот	рейс	дата	время_вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

При добавлении информации о новом рейсе 412, вылетающем в 13:25, придётся добавлять информацию и о пилоте, и о дате вылета (лишнюю для конкретного рейса)

Противодействие избыточности

- ▶ Оказывается, что такого рода ситуации достаточно часты, и для их решения был создан механизм, который называется **нормализацией**
- ▶ **Суть нормализации** состоит в следующем: представить конкретные реляционные отношения таким образом, чтобы насколько возможно избежать избыточности данных
- ▶ Для этого схему R исходного отношения r чаще всего разбивают на несколько схем, $\{R_1, \dots, R_m\}$ так что

$$R_i \subseteq R, i = \overline{1, m} \quad \text{и} \quad \bigcup_{i=1}^m R_i \supseteq R$$

Противодействие избыточности

Однако, на основании каких принципов производить разбиение схемы на подсхемы ?

Основной **постулат**, принятый при решении проблемы избыточности данных, сводится к следующему:

Чаще всего, между данными существуют зависимости

Выявление этих зависимостей – один из способов формализации семантики данных



Противодействие избыточности

- ▶ Рассмотрим тот же пример – отношение

график (пилот, рейс, дата, время_вылета),

и определим, насколько возможно, зависимости между его атрибутами



Пример избыточности хранения



график (

пилот	рейс	дата	время вылета
Кушинг	83	9.авг.	10:15
Кушинг	116	10.авг.	13:25
Кларк	281	8.авг.	5:50
Кларк	301	12.авг.	18:35
Кларк	83	11.авг.	10:15
Чин	83	13.авг.	10:15
Чин	116	12.авг.	13:25
Коупли	281	9.авг.	5:50
Коупли	281	13.авг.	5:50
Коупли	412	15.авг.	13:25

)

Для каждого рейса назначается единственное время вылета
рейс → время_вылета

Пример избыточности хранения



график (пилот	рейс	дата	время вылета)
	Кушинг	83	9.авг.	10:15	
	Кушинг	116	10.авг.	13:25	
	Кларк	281	8.авг.	5:50	
	Кларк	301	12.авг.	18:35	
	Кларк	83	11.авг.	10:15	
	Чин	83	13.авг.	10:15	
	Чин	116	12.авг.	13:25	
	Коупли	281	9.авг.	5:50	
	Коупли	281	13.авг.	5:50	
	Коупли	412	15.авг.	13:25	

Для данных пилота, даты и времени вылета возможен только один рейс

{пилот, дата, время_вылета} → рейс

Пример избыточности хранения



график (пилот	рейс	дата	время_вылета)
	Кушинг	83	9.авг.	10:15	
	Кушинг	116	10.авг.	13:25	
	Кларк	281	8.авг.	5:50	
	Кларк	301	12.авг.	18:35	
	Кларк	83	11.авг.	10:15	
	Чин	83	13.авг.	10:15	
	Чин	116	12.авг.	13:25	
	Коупли	281	9.авг.	5:50	
	Коупли	281	13.авг.	5:50	
	Коупли	412	15.авг.	13:25	

Для данного рейса и даты назначается только один пилот
{рейс, дата} → пилот

Множество F-зависимостей

- ▶ Для реляционного отношения обычно известно множество F-зависимостей:

$$S = \{ \text{рейс} \rightarrow \text{время_вылета}, \\ \{ \text{пилот}, \text{дата}, \text{время_вылета} \} \rightarrow \text{рейс}, \\ \{ \text{рейс}, \text{дата} \} \rightarrow \text{пилот} \}$$

Говорят, что

S – множество F-зависимостей, которые выполняются для отношения *график* (пилот, рейс, дата, время_вылета)



Формальные определения

Пусть r – отношение со схемой R , X и Y – подмножества R .

Отношение r удовлетворяет функциональной зависимости (F-зависимости) $X \rightarrow Y$, если

$$\forall t_1, t_2 \in r, t_1(X) = t_2(X) \Rightarrow t_1(Y) = t_2(Y)$$

X - левая часть, Y - правая часть функциональной зависимости

- ▶ Говорят еще, что на **схеме R выполняется функциональная (или F-) зависимость $X \rightarrow Y$** , поскольку нас не будут интересовать зависимости, которые могут не выполняться при другом сочетании значений кортежей

Формальные определения (2)

Функциональная зависимость является
обобщением понятия первичного ключа:

Для любого реляционного отношения $r (R)$:

$$R = A_1 \cup \dots \cup A_n \cup B_1 \cup \dots \cup B_m$$

где $A = \bigcup_{i=1}^n A_i$ - первичный ключ схемы R ,

а $B = \bigcup_{i=1}^m B_i$ - множество неключевых атрибутов,

верно, что $A \rightarrow B$, и вообще, $A \rightarrow R$ (т.е. всю схему)



Использование F-зависимостей

- 1) Любая F-зависимость является специализированным правилом целостности в реляционной модели данных

Почему ?:

Потому что любая F-зависимость $X \rightarrow Y$ накладывает ограничения на **все** допустимые значения атрибутов X и Y в отношении

А значит, знание множества F-зависимостей для каждого отношения БД - всё равно что знание множества всех ограничений целостности этой БД



F-зависимости как ограничения целостности

- ▶ При каждом обновлении БД СУБД должна проверить все ограничения целостности, в том числе и все F-зависимости между данными, затронутыми обновлением.
- ▶ Т.к. в реальной базе данных множество всех F-зависимостей может быть достаточно большим, то *желательно* для заданного множества F-зависимостей S найти такое множество T , которое:
 - Во-первых, значительно меньше множества S ;
 - Во-вторых, каждую F-зависимость из S можно заменить F-зависимостью из T .
- ▶ Тогда СУБД может использовать только F-зависимости из T – и работать быстрее

Получение минимального множества F -зависимостей

- ▶ Как найти подходящее множество T ?

Для этого:

1. Находят все возможные функциональные зависимости, исходя из множества S (замыкание множества S , S^+).
2. Находят T' -покрытие, такое, что количество функциональных зависимостей в T' меньше, чем в S .
3. Находят T – минимальное покрытие из всех найденных T'



Получение минимального множества F-зависимостей

Для нахождения всех возможных F-зависимостей для данной БД используются так называемые **аксиомы вывода Армстронга** (или правила вывода Армстронга):

Для любых подмножеств атрибутов X, Y, Z, W схемы R , где R - схема реляционного отношения, справедливы (запись « XZ » означает « $X \cup Z$ »)

I. Рефлексивность: $X \rightarrow X$

II. Пополнение: $X \rightarrow Y \Rightarrow XZ \rightarrow Y$

III. Аддитивность: $X \rightarrow Y$ и $X \rightarrow Z \Rightarrow X \rightarrow YZ$

IV. Проективность: $X \rightarrow YZ \Rightarrow X \rightarrow Y$

V. Транзитивность: $X \rightarrow Y$ и $Y \rightarrow Z \Rightarrow X \rightarrow Z$

VI. Псевдотранзитивность: $X \rightarrow Y$ и $YZ \rightarrow W \Rightarrow XZ \rightarrow W$

Формальные определения (5)

Аксиомы I, II и VI образуют полный набор правил вывода для функциональных зависимостей

Аксиомы III, IV, V можно вывести из этого минимального набора

С помощью аксиом Армстронга получают множество всех F-зависимостей для данной схемы R



Полное множество F-зависимостей

Для любого множества S функциональных зависимостей для схемы R

S^+ (замыкание S) – наименьшее содержащее S множество, такое, что, применяя к нему аксиомы Армстронга, нельзя получить ни одной новой F-зависимости, не принадлежащей S^+ .



Полное множество F-зависимостей

- ▶ Замыкание множества F-зависимостей может быть довольно большим

Пусть $S = \{ AB \rightarrow C, C \rightarrow B \}$, схема $R=(A,B,C)$

$S^+ = \{$ $A \rightarrow A(I), AB \rightarrow A(II), AC \rightarrow A(II), ABC \rightarrow A(II),$
 $B \rightarrow B(I), AB \rightarrow B(II), BC \rightarrow B(II), ABC \rightarrow B(II),$
 $C \rightarrow C(I), AC \rightarrow C(II), BC \rightarrow C(II), ABC \rightarrow C(II),$
 $AB \rightarrow AB(I), ABC \rightarrow AB(II), AC \rightarrow AC(I),$
 $ABC \rightarrow AC(III),$
 $BC \rightarrow BC(I), ABC \rightarrow BC(III), ABC \rightarrow ABC(I),$
 $AB \rightarrow C(\text{дано}), AB \rightarrow AC(III), AB \rightarrow BC(III),$
 $AB \rightarrow ABC(III),$
 $C \rightarrow B(\text{дано}), C \rightarrow BC(III), AC \rightarrow B(V),$
 $AC \rightarrow AB(III)\}$ //всего – 27 F-зависимостей

Проверка принадлежности к S^+

- ▶ Чтобы проверить, влечёт ли множество S F -зависимость $X \rightarrow Y$, достаточно проверить, принадлежит ли $X \rightarrow Y$ к S^+
- ▶ S^+ обычно велико
- ▶ Желательно найти способ проверки принадлежности $X \rightarrow Y$ к S^+ без вычисления S^+
- ▶ Это возможно
- ▶ Достаточно построить X^+ над S :
алгоритм CLOSURE (X, S),
- ▶ а потом проверить выводимость $X \rightarrow Y$ из S :
алгоритм MEMBER ($S, X \rightarrow Y$)



Алгоритм вычисления X^+ над S

Алгоритм **CLOSURE** (X, S):

Вход: множество атрибутов X , множество F -зависимостей S

Выход: замыкание X^+ над S

begin

OLDDEP:= \emptyset ; NEWDEP:= X ;

while NEWDEP \neq OLDDEP **do begin**

 OLDDEP:=NEWDEP;

for каждая F -зависимость $W \rightarrow Z$ из S **do**

if $W \subseteq$ NEWDEP **then**

 NEWDEP:= NEWDEP \cup Z

end;

return (NEWDEP);

end.

Сложность алгоритма = $O(a \cdot p^2)$, где a – количество имён атрибутов в X , p – количество F -зависимостей в S

Алгоритм проверки принадлежности к S^+

Алгоритм **MEMBER** ($S, X \rightarrow Y$):

Вход: множество F-зависимостей S , F-зависимость $X \rightarrow Y$

Выход: истина, если $S \models X \rightarrow Y$; ложь – в противном случае

begin

if $Y \subseteq \text{CLOSURE}(X, S)$

then return (истина),

else return (ложь)

end.



Полное множество F-зависимостей

Проверка факта, что некоторая F-зависимость $X \rightarrow Y$ выводится из множества S – итеративный процесс, который можно рассматривать как построение направленного ациклического графа – DA-графа.



Полное множество F-зависимостей

DA-граф над S – ориентированный ациклический граф, построенный по таким правилам:

R1. Любое множество изолированных вершин с метками из R – DA-граф.

R2. Если H – DA-граф вывода над S , содержит вершины v_1, v_2, \dots, v_k с метками A_1, \dots, A_k ,
и $\{A_1, A_2, \dots, A_k\} \rightarrow CZ$ – F-зависимость в S ,

То: граф H' строится добавлением к графу H вершины w с меткой C , и добавлением дуг $(v_1, w), (v_2, w), \dots, (v_k, w)$.

Граф H' – является DA-графом вывода над S .

R3. Никакой другой граф не является DA-графом вывода над S .



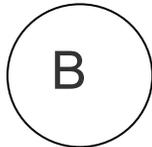
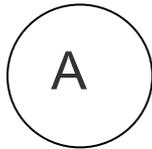
Пример DA-графа

$S = \{AB \rightarrow E, AG \rightarrow J, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

F-зависимость, принадлежность которой к S проверяется:

$AB \rightarrow GH$

Правило R1

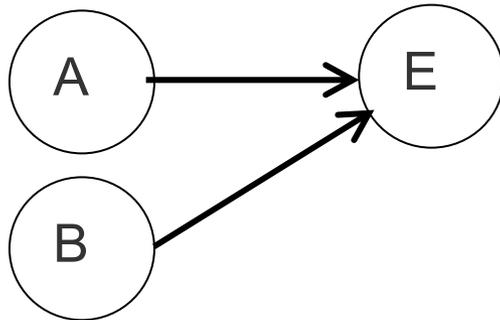


Пример DA-графа

$S = \{AB \rightarrow E, AG \rightarrow J, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

F-зависимость, принадлежность которой к S проверяется:

$AB \rightarrow GH$



Правило R2

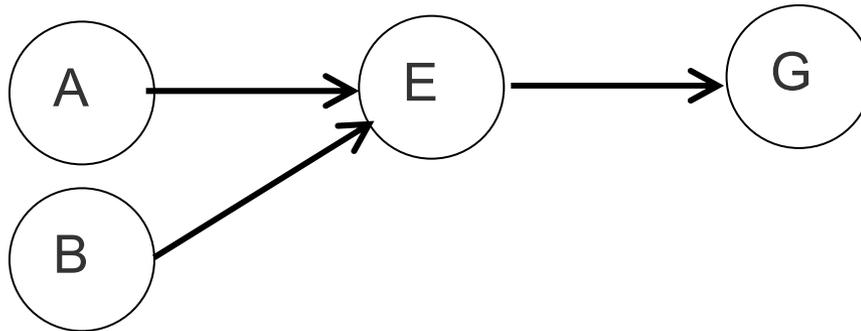
$AB \rightarrow E$



Пример DA-графа

$S = \{AB \rightarrow E, AG \rightarrow J, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

F-зависимость, принадлежность которой к S проверяется:
 $AB \rightarrow GH$



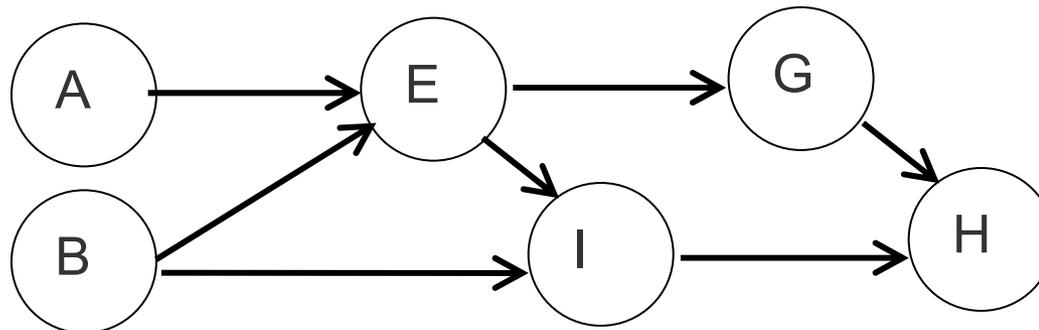
Правило R2
 $E \rightarrow G$



Пример DA-графа

$S = \{AB \rightarrow E, AG \rightarrow J, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

F-зависимость, принадлежность которой к S проверяется:
 $AB \rightarrow GH$



Правило R2

$BE \rightarrow I$

$GI \rightarrow H$

Данный граф – это DA-граф для $AB \rightarrow GH$.

Его используемое множество (множество использованных из S зависимостей):

$\{AB \rightarrow E, E \rightarrow G, BE \rightarrow I, GI \rightarrow H\}$

Проверка выводимости F-зависимости

Справедлива теорема(*):

Для данного множества F-зависимостей S над схемой R и F-зависимости $X \rightarrow Y$ следующие утверждения эквивалентны:

1. $F \models X \rightarrow Y$
2. Существует последовательность вывода на S для $X \rightarrow Y$
3. Существует DA-граф над S для $X \rightarrow Y$

(*) Д. Мейер, Теория реляционных баз данных, стр. 67



F-зависимости и поиск ключей

- ▶ Оказывается, что функциональные зависимости несут в себе еще одну полезную информацию.
- ▶ Допуская, что известны некоторые F-зависимости для конкретной схемы R, по этим F-зависимостям можно определить все потенциальные ключи для схемы R.
- ▶ Критерий для определения того, что K – ключ, простой:
 - Если в процессе построения S+ обнаруживается такое подмножество атрибутов K, что $S \models K \rightarrow R$ (где R- схема отношения) то это подмножество и есть ключ
- ▶ Таких ключей обычно бывает несколько – поэтому выбирают минимальный

Что узнали сегодня?

- ▶ **Избыточность данных** – частое явление в плохо спроектированных схемах БД
- ▶ Избыточность данных – **иногда бывает необходима** (но в таком случае существует обоснование необходимости избыточности)
- ▶ **Способ борьбы** с избыточностью – нормализация
- ▶ **Нормализация** основывается на выявлении зависимостей между атрибутами отношений схемы БД
- ▶ Наиболее известные зависимости – функциональные (**F-зависимости**) – специальные ограничения целостности
- ▶ Есть несколько задач, в которых знание F-зависимостей приносит **пользу**:
 - Получение минимального множества зависимостей (для экономии времени на проверку специальных ограничений целостности при обновлении БД)
 - Нахождение первичных ключей минимального размера