Не функциональные зависимости

СЛУЖ_ПРО_ЗАДАН

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	A
34	1	B
34	2	A
34	2	B
41	1	A
41	1	D

- служащий может участвовать в нескольких проектах
- в каждом проекте служащий выполняет одинаковый набор заданий

Ключ отношения {СЛУ_N, ПРО_N, СЛУ_ЗАДАН}
Находится в BCBF (отсутствуют нетривиальные FD)

IF(<с,п1,з1>∈СПЗ AND <с,п2,з2>∈СПЗ)THEN(<с,п1,з2>∈СПЗ AND <с,п2,з1>∈СПЗ)

Аномалии обновления:

- **Добавление кортежа**. Если служащий присоединяется к проекту, то необходимо добавить столько кортежей, сколько заданий выполняет этот служащий.
- **Удаление кортежей**. Если служащий прекращает участие в проектах, то отсутствует возможность сохранить данные о заданиях, которые он может выполнять.
- **Модификация кортежей**. При изменении одного из заданий служащего необходимо изменить значение атрибута СЛУ_ЗАДАН в нескольких кортежах, для каждого проекта.

Возможная декомпозиция:

СЛУЖ_ПРО

СЛУ_N	ПРО_N
34	1
34	2
41	1

СЛУЖ_ЗАДАН

СЛУ_N	СЛУ_ЗАДАН
34	A
34	B
41	A
41	D

В отношении нет FD, однако:

1. Декомпозиция без потерь
2. Проблемы с аномалиями решены

Многозначная зависимость.

Впервые описаны Роном Фейджином в 1971 г. Фейджин назвал зависимости этого вида многозначными (*multi-valued dependency* – MVD).

В СЛУЖ_ПРО_ЗАДАН выполняются две MVD: СЛУ_N→→ПРО_N и СЛУ_N→→СЛУ_ЗАДАН

В терминах алгебраических выражений MVD СЛУ_N→→ПРО_N означает, что результат:
Ri = (СЛУЖ_ПРО WHERE (СЛУ_N = ‘N’ AND СЛУ_ЗАДАН = ‘Zi’)) PROJECT {ПРО_N}
‘N’ = const & ∀ Zi, Zj => Ri = Rj

В переменной отношения r с атрибутами A, B, C (в общем случае, составными) имеется **многозначная зависимость** B от A (A→→B) в том и только в том случае, когда множество значений атрибута B, соответствующее паре значений атрибутов A и C, зависит от значения A и не зависит от значения C.

IF (<a, b, c’> AND <a, b’, c>) THEN (<a, b, c>)

В силу симметрии атрибутов мы можем также записать правило в виде:

IF (<a, b, c’> AND <a, b’, c>) THEN (<a, b, c> AND <a, b’, c’>)

Лемма Фейджина о «двойственности» многозначной зависимости:

В отношении r {A, B, C} выполняется MVD A→→B в том и только в том случае, когда выполняется MVD A→→C.

Доказательство достаточности условия леммы.
Пусть ∃ MVD A→→B.
r - отношение,
∀ a : a (A ∈ Hr) ∈ t - значение атрибута A ∈ Hr в некотором кортеже тела r,
{b} – множество значений атрибута B ∈ Hr, взятых из всех кортежей тела Vr, в которых значением атрибута A является a.
Предположим, что для этого значения a MVD A→→C не выполняется.
Это означает, что существуют такое допустимое значение c атрибута C и такое значение b∈{b}, что кортеж <a, b, c>∉Vr.
Но это противоречит наличию MVD A→→B. Следовательно, если выполняется MVD A→→B, то выполняется и MVD A→→C.
Необходимость условия леммы доказывается симметрично.

MVD A→→B и A→→C представляют в форме A→→B|C

Многозначные и функциональные зависимости

FD - частный случай MVD, когда множество значений зависимого атрибута состоит из одного элемента.

если $\exists$ FD $A \rightarrow B$ (r,Hr{A,B,C}), то => $\exists$ MVD $A \rightarrow\rightarrow B|C$

Если <a,b,c1> $\in$ r AND <a,b1,c> $\in$ r, тогда в силу $A \rightarrow B$ =>
b1 = b, то есть <a,b1,c> = <a,b,c> =>
<a,b,c> $\in$ r =>
$\exists$ $A \rightarrow\rightarrow B|C$ по определению.

Обратное – не верно.
Многозначная зависимость является обобщением понятия функциональной зависимости.

Теорема Фейджина о декомпозиции отношения с многозначной зависимостью

r отношение на Hr {A, B, C}
Отношение r декомпозируется без потерь на проекции {A, B} и {A, C} тогда и только тогда, когда для него выполняется MVD A→→B|C.

r = r1 NATURALJOIN r2, r1=r PROJECT{AB}, r2=r PROJECT{AC} ⇔ MVD A→→B|C

[согласно определению MVD A→→B : IF (<ab1c> ∈ r AND <abc1> ∈ r) THEN (<abc> ∈ r)]

Доказательство **необходимости** условия теоремы (т.е. ⇒)
Левая часть выполняется, т.е. r = r1 NATURALJOIN r2 (декомпозиция без потерь)
Докажем что в таком случае выполняется MVD A→→B |C.
Пусть <abc1> ∈ r и <ab1c> ∈ r =>
По определению проекций <ab> ∈ r1, <ac> ∈ r2 =>
<abc> ∈ r1 NATURALJOIN r2, если исходная декомпозиция – без потерь =>
<abc> ∈ r – а это и есть определение MVD A→→B.
Чтд, необходимость доказана.

Доказательство **достаточности** условия теоремы (т.е. ⇐).
Пусть имеется многозначная зависимость A→→B|C.
Докажем, что декомпозиция r = r1 NATURALJOIN r2 является декомпозицией без потерь.
Доказательство аналогично теореме Хитта и разбито на две части.
Первая – включение r ⊆ r1 NATURALJOIN r2 – то есть надо доказать что любой кортеж исходного отношения принадлежит результату - доказывается так-же как у Хитта на основе определения операций проекции и естественного соединения и выполняется всегда для любого состояния r.
Докажем вторую половину, а именно что r1 NATURALJOIN r2 ⊆ r.
Пусть <abc> ∈ r1 NATURAL JOIN r2 **=>** <ab> ∈ r1, <ac> ∈ r2 **=>** <abc1> ∈ r и <ab1c> ∈ r **=>**
по определению многозначной зависимости => <abc> ∈ r.
Включение доказано. **Достаточность доказана**. Теорема Фейджина полностью доказана.

Четвертая нормальная форма (4NF)

Переменная отношения r находится в **четвертой нормальной форме** (**4NF**) в том и только в том случае, когда она находится в BCNF, и все MVD r являются FD с детерминантами – возможными ключами отношения r.

4NF является BCNF, в которой многозначные зависимости вырождаются в функциональные.

СЛУЖ_ПРО_ЗАДАН не находится в 4NF, поскольку MVD СЛУ_N→→ПРО_N |СЛУ_ЗАДАН не являются функциональными и детерминант не является возможным ключом отношения.

СЛУЖ_ПРО_N и СЛУЖ_ЗАДАНИЕ находятся в BCNF поскольку не содержат MVD, отличных от FD с детерминантом – возможным ключом (нет нетривиальных FD). Поэтому они находятся в 4NF.

Многозначная зависимость MVD A→→B|C называется **нетривиальной многозначной зависимостью**, если не существует функциональных зависимостей FD A→B или FD A→C.

СЛУ_N→→ПРО_N|СЛУ_ЗАДАН – пример нетривиальной многозначной зависимостью.

MVD A→→B называется **тривиальной** если либо B $\in$ A, либо A $\cup$ B = Hr.
Тривиальная MVD всегда удовлетворяется.
При B $\in$ A она вырождается в тривиальную FD.
При A $\cup$ B = Hr MVD верна по определению.

N-декомпозируемые отношения

Будем называть N-**декомпозируемым** отношением отношение, которое может быть декомпозировано без потерь на N проекций.

СЛУЖ_ПРО_ЗАДАН (ключ Hr и отсутствуют нетривиальные MVD)

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	A
34	1	B
34	2	A
41	1	A

СЛУЖ_ПРО

СЛУ_N	ПРО_N
34	1
34	2
41	1

ПРО_ЗАДАН

ПРО_N	СЛУ_ЗАДАН
1	A
1	B
2	A

СЛУЖ_ЗАДАН

СЛУ_N	СЛУ_ЗАДАН
34	A
34	B
41	A

СЛУЖ_ПРО NATURAL JOIN ПРО_ЗАДАН

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	A
34	1	B
34	2	A
41	1	A
41	1	B

r != r1 NATURALJOIN r2
r != r2 NATURALJOIN r3
r != r1 NATURALJOIN r3

=> по Фейджину – нетривиальные MVD отстутствуют

СЛУЖ_ПРО NATURAL JOIN СЛУЖ_ЗАДАН
<34,2,B> - лишний
ПРО_ЗАДАН NATURAL JOIN СЛУЖ_ЗАДАН
<41,2,A> - лишний

НО
r = r1 NATURALJOIN r2 NATURALJOIN r3

Это говорит о том, что между атрибутами этого отношения имеется некоторая зависимость, но эта зависимость не является ни функциональной, ни многозначной зависимостью.

Зависимость проекции/соединения

Обозначим СПЗ = СЛУЖ_ПРО_ЗАДАН, СП = СЛУЖ_ПРО, ПЗ = ПРО_ЗАДАН, СЗ = СЛУЖ_ЗАДАН

СПЗ = СП NATURAL JOIN ПЗ NATURAL JOIN СЗ
⇔
IF (<сп> ∈ СП AND <пз> ∈ ПЗ AND <сз> ∈ СЗ) THEN <спз> ∈ СПЗ

Ограничение, обеспечивающее возможность восстановления без потерь, определенное на исходном отношении будет выглядеть следующим образом:

IF (<спз'> ∈ СПЗ AND <сп'з> ∈ СПЗ AND <с'пз> ∈ СПЗ) THEN <спз> ∈ СПЗ

Если служащий «С» участвует в проекте «П» выполняя какое-либо задание,
и в проекте «П» выполняется задание «З» каким-либо служащим,
и служащий «С» выполняет задание «З» в каком-либо проекте,
ТО служащий «С» выполняет задание «З» в проекте «П».

Зависимость проекции/соединения (**Project-Join Dependency – PJD**)

Отношение r на Hr{A,B,…Z} (составные, перекрывающиеся атрибуты).
В переменной отношения r удовлетворяется **зависимость проекции/соединения *(AB… Z)** тогда и только тогда, когда любое допустимое значение r можно получить путем естественного соединения проекций этого значения на атрибуты AB…Z.
r = r{A} NATURALJOIN r{B} NATURALJOIN … NATURALJOIN r{Z}
При этом данная зависимость должна выполняться для любого значения тела отношения, удовлетворяющего всем наложенным на него ограничениям предметной области.

Аномалии, вызываемые наличием зависимости проекции/соединения

В СПЗ_2 ∃ PJD *({СЛУ_N, ПРО_N}, {ПРО_N, СЛУ_ЗАДАН}, {СЛУ_N, СЛУ_ЗАДАН})

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	B
34	2	A

IF (<спз'> ∈ СПЗ AND <сп'з> ∈ СПЗ AND <с'пз> ∈ СПЗ) THEN <спз> ∈ СПЗ

- **Добавление кортежей**.
  СПЗ_2 добавить <41,1,A>,

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	B
34	2	A
41	1	A

НЕ УДОВЛЕТВОРЯЕТ ограничению!

34	1	A

  Кортежи <34,1,B>, <34,2,A> и <41,1,A> дают нам пары 34-1, 34-A, 1-A =>
  Ограничение целостности требует включения в отношение картежа <34,1,A>
  [добавление <34,1,A> не нарушает ограничения и не требует других добавлений]
- **Удаление кортежа**.
  Если из СПЗ_2 удаляется кортеж <34,1,A>, то нужно удалить и кортеж <41,1,A>. Поскольку в соответствии с ограничением целостности наличие второго кортежа означает наличие первого.
  [удаление <41,1,A> не нарушает ограничения и не требует других удалений]

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	B
34	2	A

СЛУЖ_ПРО

СЛУ_N	ПРО_N
34	1
34	2

ПРО_ЗАДАН

ПРО_N	СЛУ_ЗАДАН
1	B
2	A

СЛУЖ_ЗАДАН

СЛУ_N	СЛУ_ЗАДАН
34	A
34	B

+ <41,1,A>

СЛУЖ_ПРО

СЛУ_N	ПРО_N
34	1
34	2
41	1

ПРО_ЗАДАН

ПРО_N	СЛУ_ЗАДАН
1	B
2	A
1	A

СЛУЖ_ЗАДАН

СЛУ_N	СЛУ_ЗАДАН
34	A
34	B
41	A

СЛУЖ_ПРО_ЗАДАН = СЛУЖ_ПРО NATURAL JOIN ПРО_ЗАДАН NATURAL JOIN СЛУЖ_ЗАДАН

СЛУ_N	ПРО_N	СЛУ_ЗАДАН
34	1	A
34	1	B
34	2	A
41	1	A

Связь многозначной зависимости и зависимости проекции/соединения

Зависимость проекции/соединения фактически является обобщением понятия многозначной зависимости, введенной нами для четвертой нормальной формы.

Теорема Фейджина (формулировка для зависимости проекции/соединения).

Отношение r(ABC) удовлетворяет зависимости проекции/соединения *(AB, AC) тогда и только тогда, когда имеется многозначная зависимость A->>B|C.

r(ABC) == (r PROJECT(AB)) NATURAL JOIN (r PROJECT (AC)) ⇔ *(AB,AC) ⇔ A->>B|C

Многозначная зависимость - частный случай зависимости проекции-соединения, т.е., если в отношении имеется многозначная зависимость, то имеется и зависимость проекции-соединения на 2 соответствующих атрибута.

Обратное (для любого количества атрибутов), неверно.

Тривиальные и нетривиальные PJD

Зависимость проекции/соединения *(AB…Z) в отношении r называется **подразумеваемой возможными ключами** в том и только в том случае, когда каждый составной атрибут AB…Z является суперключом r, т.е. включает хотя бы один возможный ключ r.

Зависимость проекции/соединения *(AB…Z) в отношении r называется **тривиальной**, если хотя бы один из составных атрибутов AB…Z совпадает с заголовком r.

Нетривиальные PJD, подразумеваемые возможными ключами, существуют во всех отношениях с арностью, большей двух, первичный ключ которых не совпадает с заголовком отношения.

Примеры:
- СЛУЖ_ПРО_ЗАДАН (СЛУ_N - первичный ключ)
  (служащий может работать только в одном проекте и над одним заданием)
  то ∃ PJD *({СЛУ_N, ПРО_N}, {СЛУ_N, СЛУ_ЗАДАН})
- r(ABCD) (A – первичный ключ == A->B,A->C,A->D == PR(AB), PR(AC), PR(AD))
  то ∃ PJD *({AB},{AC},{AD})

Нетривиальные PJD, подразумеваемые возможными ключами неинтересны с точки зрения проектирования базы данных, поскольку они не порождают аномалии обновления.

Пятая нормальная форма

Отношение r находится в **пятой нормальной форме**, или в **нормальной форме проекции/соединения** (**5NF**, **PJ/NF, Project-Join Normal Form**) в том и только в том случае, когда каждая нетривиальная PJD в r подразумевается возможными ключами r.

5NF является «окончательной» нормальной формой, которой можно достичь в процессе нормализации на основе проекций.
У отношения, находящегося в 5NF, отсутствуют аномалии обновлений, которые можно было бы устранить путем его декомпозиции.
Дальнейшая нормализация отношения в 5NF бессмысленна.

Актуальность процесса нормализации

Первый аспект: Реляционная модель данных != SQL модель данных
Подавляющее большинство современных баз данных и средств управления ими опирается на модель данных SQL, которая не тождественна реляционной модели.

(*например, таблица модели SQL может содержать мультимножества строк и не иметь ключа*).

Но если потребовать от проектируемой SQL-базы данных наличия хотя бы одного возможного ключа для каждой таблицы целевой базы данных, то все методы нормализации становятся применимы.
Таким образом, SQL-ориентированную базу данных можно проектировать как реляционную базу данных, если должным образом ограничить используемые средства модели данных SQL.

В дальнейшем под «**реляционными**» базами данных будут пониматься именно SQL-ориентированные базы данных, не противоречащие требованиям реляционной модели данных.

Актуальность процесса нормализации

Второй аспект: актуальность «нормализации»

Основные цели процесса нормализации:

- избежать избыточности хранения данных;
- устранить аномалии обновления отношений.

Хорошо нормализованные реляционные базы данных в значительной степени способствуют росту эффективности приложений. И на первых шагах развития это было особенно актуально.

Теория реляционных баз данных и методы их проектирования активно развиваются уже более 25 лет. И ситуация и в области аппаратного и в области программного обеспечения не стоит на месте.

Однако и сейчас наиболее частой схемой использования СУБД остается работа в информационных системах оперативной обработки транзакций
(On-Line Transaction Processing – OLTP).

Информационные системы оперативной обработки транзакций
On-Line Transaction Processing – OLTP

операционные банковские системы,
системы учета,
системы заказа и резервирования,
и многие другие.

Общая характеристика OLTP систем

- количество транзакций очень велико,
- выполняются транзакции параллельно,
- сложность транзакций низкая,
- необходима высокая эффективность отката транзакции,
- транзакции на вставку/удаления/модификацию превышают запросы на выборку,
- большая часть планируемых запросов известна еще на стадии проектирования.

Вывод:
чем выше уровень нормализации данных в OLTP-приложениях, тем они быстрее и надежней. Возможны отступления для обеспечения эффективности заранее известных запросов, которые критичны для работы приложений.

Системы категории оперативной аналитической обработки
On-Line Analytical Processing – OLAP

OLAP - обобщенный термин, характеризующий принципы построения систем, предназначенных для нахождения зависимостей между данными, для проведения динамического анализа данных и тому подобных задач.
К **OLAP** системам можно отнести системы:
- поддержки принятия решений – Decision Support System (DSS),
- хранилища данных – Data Warehouse,
- системы интеллектуального анализа данных – Data Mining,
- …

OLAP-приложения оперируют с большими массивами данных и характеризуются следующими признаками:

- добавление в систему новых данных происходит относительно редко крупными блоками;
- данные, добавленные в систему, как правило, никогда не удаляются;
- перед загрузкой данные могут проходить подготовительную обработку;
- запросы к системе являются нерегламентированными и достаточно сложными;
- скорость выполнения запросов важна, но не столь критична как в OLTP системах.

Базы данных OLAP-приложений обычно представлены в виде **гиперкубов -** конструкций, каждое измерение которой представляет собой некоторую характеристику данные, а в ячейках самого гиперкуба хранятся значения этих данных.

Физически гиперкуб может быть построен на основе специальной многомерной модели данных – **Multidimensional OLAP** (**MOLAP**) или средствами реляционной модели данных – **Relational OLAP** (**ROLAP**).

В системах OLAP, использующих реляционную модель данных, данные целесообразно хранить в виде слабо нормализованных отношений, содержащих заранее вычисленные основные итоговые данные. Аномалии обновления мало влияют на работу систем.

Технологически правильным считается для OLAP систем поддерживать отдельную базу данных. Источником данных для такой системы обычно служат различные OLTP системы.

Эффективность работы таких систем обычно сильно зависит от того насколько правильно они спроектированы. Обычно первым шагом проектирования проводят процедуру нормализации схемы системы до максимально возможного уровня нормализации, а вторым шагом проводят процесс денормализации из соображения повышения эффективности выполнения запросов.