

КОНСПЕКТ ЗАНЯТИЯ № 1 ПЯТОЙ НЕДЕЛИ КУРСА «БАЗЫ ДАННЫХ»

1. НОРМАЛЬНЫЕ ФОРМЫ БОЛЕЕ ВЫСОКОГО ПОРЯДКА

На предыдущем занятии были рассмотрены нормальные формы вплоть до третьей нормальной формы (3НФ). В большинстве случаев этого вполне достаточно, чтобы разрабатывать вполне работоспособные базы данных. В данной главе рассматриваются нормальные формы более высоких порядков, а именно, нормальная форма Бойса-Кодда (НФБК), четвертая нормальная форма (4НФ) и пятая нормальная форма (5НФ).

1.1. НФБК (Нормальная форма Бойса-Кодда)

При приведении отношений при помощи алгоритма нормализации к отношениям в 3НФ неявно предполагалось, что все отношения содержат *один* потенциальный ключ. Это не всегда верно. Рассмотрим следующий пример отношения, содержащего два ключа.

Пусть требуется хранить данные о поставках деталей некоторыми поставщиками. Предположим, что наименования поставщиков являются уникальными. Кроме того, каждый поставщик имеет свой уникальный номер. Данные о поставках можно хранить в следующем отношении:

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Поставляемое количество VOLUME
1	Фирма 1	1	100
1	Фирма 1	2	200
1	Фирма 1	3	300
2	Фирма 2	1	150
2	Фирма 2	2	250
3	Фирма 3	1	1000

Таблица 1 Отношение "Поставки"

Данное отношение содержит два потенциальных ключа - **{PNUM, DNUM}** и **{PNAME, DNUM}**. Видно, что данные хранятся в отношении с избыточностью - при изменении наименования поставщика, это наименование нужно изменить во всех кортежах, где оно встречается. Можно ли эту аномалию устранить при помощи алгоритма нормализации,

описанного в предыдущей главе? Для этого нужно выявить имеющиеся функциональные зависимости (курсивом выделены ключевые атрибуты):

PNUM → *PNAME* - наименование поставщика зависит от номера поставщика.

PNAME → *PNUM* - номер поставщика зависит от наименования поставщика.

{*PNUM*, *DNUM*} → *VOLUME* - поставляемое количество зависит от первого ключа отношения.

{*PNUM*, *DNUM*} → *PNAME* - наименование поставщика зависит от первого ключа отношения.

{*PNAME*, *DNUM*} → *VOLUME* - поставляемое количество зависит от второго ключа отношения.

{*PNAME*, *DNUM*} → *PNUM* - номер поставщика зависит от второго ключа отношения.

Данное отношение не содержит неключевых атрибутов, зависящих от части сложного ключа (см. определение 2НФ). Действительно, от части сложного ключа зависят атрибуты *PNAME* и *PNUM*, но они сами являются ключевыми. Таким образом, отношение находится в 2НФ.

Кроме того, отношение не содержит зависимых друг от друга неключевых атрибутов, т.к. неключевой атрибут всего один - *VOLUME* (см. определение 3НФ). То есть, отношение "Поставки" находится в 3НФ.

Таким образом, описанный ранее алгоритм нормализации неприменим к данному отношению. Очевидно, однако, что аномалия данного отношения устраняется путем декомпозиции его на два следующих отношения:

Номер поставщика <i>PNUM</i>	Наименование поставщика <i>PNAME</i>
1	Фирма 1
2	Фирма 2
3	Фирма 3

Таблица 2 Отношение "Поставщики"

Номер поставщика <i>PNUM</i>	Номер детали <i>DNUM</i>	Поставляемое количество <i>VOLUME</i>
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	1	1000

Таблица 3 Отношение "Поставки-2"

Определение 1. Отношение R находится в **нормальной форме Бойса-Кодда (НФБК)** тогда и только тогда, когда *детерминанты всех функциональных зависимостей являются потенциальными ключами.*

Отношение "Поставки" не находится в НФБК, т.к. имеются зависимости ($PNUM \rightarrow PNAME$ и $PNAME \rightarrow PNUM$), детерминанты которых не являются потенциальными ключами.

Для того чтобы устранить зависимость от детерминантов, не являющихся потенциальными ключами, необходимо провести декомпозицию, вынося эти детерминанты и зависимые от них части в отдельное отношение. Отношения "Поставщики" и "Поставки-2", полученные в результате декомпозиции находятся в НФБК.

Замечание. Приведенная декомпозиция отношения "Поставки" на отношения "Поставщики" и "Поставки-2" не является единственно возможной. Альтернативной декомпозицией является декомпозиция на следующие отношения:

Номер поставщика PNUM	Наименование поставщика PNAME
1	Фирма 1
2	Фирма 2
3	Фирма 3

Таблица 4 Отношение "Поставщики"

Наименование поставщика PNAME	Номер детали DNUM	Поставляемое количество VOLUME
Фирма 1	1	100
Фирма 1	2	200
Фирма 1	3	300
Фирма 2	1	150
Фирма 2	2	250
Фирма 3	1	1000

Таблица 5 Отношение "Поставки-3"

На первый взгляд, такая декомпозиция хуже, чем первая. Действительно, наименования поставщиков по-прежнему повторяются, и при изменении наименования поставщика, это наименование придется менять одновременно в нескольких местах (тем более сразу в двух отношениях!). Кажется, что ситуация стала еще хуже, чем была до

декомпозиции. Однако такое ощущение возникает от того, что мы интуитивно считаем, что наименования поставщиков могут меняться, а номера - нет. Если же предположить, что номера поставщиков тоже могут меняться (почему бы нет - директор приказал перенумеровать поставщиков!), то первая декомпозиция получается такой же "плохой" как и вторая - повторяющиеся номера придется менять одновременно в нескольких местах и также сразу в двух отношениях.

На самом деле никакого противоречия тут нет. В отношении "Поставки-3" атрибут "Наименование поставщика" (PNAME) является *внешним ключом*, служащим для связи с отношением "Поставщики". Поэтому, при изменении наименования поставщика, это изменение производится в отношении "Поставщики" и каскадно распространяется на отношение "Поставки-3" совершенно так, как изменение номера поставщика каскадно распространяется на отношение "Поставки-2". Поэтому, формально обе декомпозиции совершенно равноправны. В реальной работе разработчик выберет, конечно, первую декомпозицию, но тут важно подчеркнуть, что его выбор основан совсем на других соображениях, не имеющих отношения к формальной теории нормальных форм.

Рассмотрим нетривиальный пример.

Предположим, что нам по-прежнему необходимо учитывать поставки, но каждый акт поставки должен иметь некоторый уникальный номер (назовем его "сквозной номер поставки"). Отношение может иметь следующий вид:

Номер поставщика PNUM	Номер детали DNUM	Поставляемое количество VOLUME	Сквозной номер поставки NN
1	1	100	1
1	2	200	2
1	3	300	3
2	1	150	4
2	2	250	5
3	1	1000	6

Таблица 6 Отношение "Поставки-с-номером"

Одним потенциальным ключом данного отношения является, как и раньше, пара атрибутов **{PNUM, DNUM}**. Другим ключом, в силу

уникальности сквозного номера, является атрибут **NN**. В данном отношении имеются следующие функциональные зависимости:

Зависимость атрибутов от первого ключа отношения:

$\{PNUM, DNUM\} \rightarrow VOLUME$,

$\{PNUM, DNUM\} \rightarrow NN$,

Зависимость атрибутов от второго ключа отношения:

$NN \rightarrow PNUM$,

$NN \rightarrow DNUM$,

$NN \rightarrow VOLUME$,

Зависимости, являющиеся следствием зависимостей от ключей отношения:

$\{PNUM, DNUM\} \rightarrow \{VOLUME, NN\}$,

$NN \rightarrow \{PNUM, DNUM\}$,

$NN \rightarrow \{PNUM, VOLUME\}$,

$NN \rightarrow \{DNUM, VOLUME\}$,

$NN \rightarrow \{PNUM, DNUM, VOLUME\}$.

Как можно заметить, детерминанты всех зависимостей являются потенциальными ключами, поэтому данное отношение находится в НФБК. Особенностью данного отношения является то, что оно имеет два совершенно независимых потенциальных ключа.

1.2. ЧЕТВЕРТАЯ НОРМАЛЬНАЯ ФОРМА (4НФ)

Рассмотрим следующий пример. Пусть требуется учитывать данные об абитуриентах, поступающих в ВУЗ. При анализе предметной области были выделены следующие требования:

- Каждый абитуриент имеет право сдавать экзамены на несколько факультетов одновременно.
- Каждый факультет имеет свой список сдаваемых предметов.
- Один и тот же предмет может сдаваться на нескольких факультетах.
- Абитуриент обязан сдавать все предметы, указанные для факультета, на который он поступает, несмотря на то, что он, может быть, уже сдавал такие же предметы на другом факультете.

Предположим, что нам требуется хранить данные о том, какие предметы должен сдавать каждый абитуриент. Попытаемся хранить данные в одном отношении "Абитуриенты-Факультеты-Предметы":

Абитуриент	Факультет	Предмет
Иванов	Математический	Математика
Иванов	Математический	Информатика
Иванов	Физический	Математика
Иванов	Физический	Физика
Петров	Математический	Математика
Петров	Математический	Информатика

Таблица 7 Отношение "Абитуриенты-Факультеты-Предметы"

В данный момент в отношении хранится информация о том, что абитуриент Иванов поступает на два факультета (математический и физический), а абитуриент Петров - только на математический. Кроме того, можно сделать вывод, что на математическом факультете нужно сдавать математику и информатику, а на физическом - математику и физику.

Аномалия обновления легко устраняется стандартным способом - вынесением всех наименований в отдельные отношения, оставляя в исходном отношении только соответствующие номера:

Номер Абитуриента	Номер Факультета	Номер Предмета
1	1	1
1	1	2
1	2	1
1	2	3
2	1	1
2	1	2

Таблица 8 Модифицированное отношение "Абитуриенты-Факультеты-Предметы"

Номер Абитуриента	Абитуриент
1	Иванов
2	Петров

Таблица 9 Отношение "Абитуриенты"

Номер Факультета	Факультет
1	Математический
2	Физический

Таблица 10 Отношение "Факультеты"

Номер Предмета	Предмет
1	Математика
2	Информатика
3	Физика

Таблица 11 Отношение "Предметы"

Теперь каждое наименование встречается только в одном месте.

И все-таки как в исходном, так и в модифицированном отношении имеются аномалии модификации, возникающие при попытке вставить или удалить кортежи.

Аномалия вставки. При попытке добавить в отношение "Абитуриенты-Факультеты-Предметы" новый кортеж, например (Сидоров, Математический, Математика), мы *обязаны* добавить также и кортеж (Сидоров, Математический, Информатика), т.к. все абитуриенты математического факультета *обязаны* иметь один и тот же список сдаваемых предметов. Соответственно, при попытке вставить в модифицированное отношении кортеж (3, 1, 1), мы обязаны вставить в него также и кортеж (3, 1, 2).

Аномалия удаления. При попытке удалить кортеж (Иванов, Математический, Математика), мы *обязаны* удалить также и кортеж (Иванов, Математический, Информатика) по той же самой причине.

Таким образом, вставка и удаление кортежей не может быть выполнена независимо от других кортежей отношения.

Кроме того, если мы удалим кортеж (Иванов, Физический, Математика), а вместе с ним и кортеж (Иванов, Физический, Физика), то будет потеряна информация о предметах, которые должны сдаваться на физическом факультете.

Декомпозиция отношения "Абитуриенты-Факультеты-Предметы" для устранения указанных аномалий не может быть выполнена на основе функциональных зависимостей, т.к. это отношение *не содержит никаких функциональных зависимостей*. Это отношение является полностью ключевым, т.е. ключом отношения является все множество атрибутов. Но ясно, что какая-то взаимосвязь между атрибутами имеется. Эта взаимосвязь описывается понятием *многозначной зависимости*.

Определение 2. Пусть R - отношение, и X, Y, Z - некоторые из его атрибутов (или *непересекающиеся* множества атрибутов).

Тогда атрибуты (множества атрибутов) Y и Z **многозначно зависят** от X (обозначается $X \twoheadrightarrow Y|Z$), тогда и только тогда, когда из того, что в отношении R содержатся кортежи $r_1 = (x, y, z_1)$ и $r_2 = (x, y_1, z)$ следует, что в отношении R содержится также и кортеж $r_3 = (x, y, z)$.

В отношении "Абитуриенты-Факультеты-Предметы" имеется многозначная зависимость Факультет \twoheadrightarrow Абитуриент|Предмет.

Словами это можно выразить так - для каждого факультета (для каждого значения из X) каждый поступающий на него абитуриент (значение из Y) сдает *один и тот же список* предметов (набор значений из Z), и для каждого факультета (для каждого значения из X) каждый сдаваемый на факультете экзамен (значение из Z) сдается *одним и тем же списком* абитуриентов (набор значений из Y). Именно наличие этой зависимости не позволяет независимо вставлять и удалять кортежи. Кортежи обязаны вставляться и удаляться одновременно *целыми наборами*.

Замечание. Если в отношении R имеется не менее трех атрибутов X, Y, Z и есть *функциональная* зависимость $X \rightarrow Y$, то есть и *многозначная* зависимость $X \twoheadrightarrow Y|Z$.

Таким образом, *понятие многозначной зависимости является обобщением понятия функциональной зависимости.*

Определение 3. Многозначная зависимость $X \twoheadrightarrow Y|Z$ называется **нетривиальной многозначной зависимостью**, если не существует функциональных зависимостей $X \rightarrow Y$ и $X \rightarrow Z$.

В отношении "Абитуриенты-Факультеты-Предметы" имеется именно нетривиальная многозначная зависимость Факультет \twoheadrightarrow Абитуриент|Предмет.

Определение 4. Отношение R находится в **четвертой нормальной форме (4НФ)** тогда и только тогда, когда отношение

находится в НФБК и *не содержит нетривиальных многозначных зависимостей*.

Отношение "Абитуриенты-Факультеты-Предметы" находится в НФБК, но не в 4НФ. Это отношение можно без потерь декомпозировать на отношения:

Факультет	Абитуриент
Математический	Иванов
Физический	Иванов
Математический	Петров

Таблица 12 Отношение "Факультеты-Абитуриенты"

Факультет	Предмет
Математический	Математика
Математический	Информатика
Физический	Математика
Физический	Физика

Таблица 13 Отношение "Факультеты-Предметы"

В полученных отношениях устранены аномалии вставки и удаления, характерные для отношения "Абитуриенты-Факультеты-Предметы".

Заметим, что полученные отношения остались полностью ключевыми, и в них по-прежнему нет функциональных зависимостей.

Отношения с нетривиальными многозначными зависимостями возникают, как правило, в результате естественного соединения двух отношений по общему полю, которое *не является ключевым ни в одном из отношений*. Фактически это приводит к попытке хранить в одном отношении информацию о двух *независимых* сущностях.

1.3. ПЯТАЯ НОРМАЛЬНАЯ ФОРМА (5НФ)

Функциональные и многозначные зависимости позволяют произвести декомпозицию исходного отношения без потерь на *две* проекции. Однако, в ряде случаев, в реляционных отношениях могут присутствовать более сложные зависимости.

Приведем пример отношения, которое нельзя декомпозировать без потерь ни на какие две проекции (отношение, для которого нельзя провести декомпозицию без потерь на 2 проекции, но можно провести декомпозицию на *n* проекций, называется *n-декомпозируемым*).

Нам дано следующее отношение R:

X	Y	Z
1	1	2
1	2	1
2	1	1
1	1	1

Таблица 14 Отношение R

Всевозможные проекции отношения R , включающие по два атрибута, имеют вид:

X	Y
1	1
1	2
2	1

Таблица 15 Проекция $R_1=R[X,Y]$

X	Z
1	2
1	1
2	1

Таблица 16 Проекция $R_2=R[X,Z]$

Y	Z
1	2
2	1
1	1

Таблица 17 Проекция $R_3=R[Y,Z]$

Как легко заметить, отношение R не восстанавливается ни по одному из *попарных* соединений $R_1 \text{ JOIN } R_2$, $R_1 \text{ JOIN } R_3$ или $R_2 \text{ JOIN } R_3$. Действительно, соединение $R_1 \text{ JOIN } R_2$ имеет вид:

X	Y	Z
1	1	2
1	1	1
1	2	2
1	2	1
2	1	1

Таблица 18 $R_1 \text{ JOIN } R_2$

В данном отношении есть лишний кортеж, отсутствующий в отношении R . Аналогично (в силу соображений симметрии) и другие попарные соединения не восстанавливают отношения R .

Однако отношение R восстанавливается соединением всех трех проекций:

$$R_1 \text{ JOIN } R_2 \text{ JOIN } R_3 = R$$

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

Определение 5. Пусть R является отношением, а A, B, \dots, Z - произвольными (возможно пересекающимися) подмножествами множества атрибутов отношения R . Тогда отношение R удовлетворяет **зависимости соединения**

$$*(A, B, \dots, Z)$$

тогда и только тогда, когда оно равносильно соединению всех своих проекций с подмножествами атрибутов A, B, \dots, Z , т.е.

$$R = R[A] \text{ JOIN } R[B] \text{ JOIN } \dots \text{ JOIN } R[Z]$$

Можно *предположить*, что отношение R в примере 3 удовлетворяет следующей зависимости соединения:

$$*(XY, XZ, YZ)$$

Утверждать, что это именно так мы пока не можем, т.к. определение зависимости соединения должно выполняться для *любого* состояния отношения R , а не только для состояния, приведенного в примере.

Зависимость соединения является обобщением понятия многозначной зависимости.

Многозначная зависимость является частным случаем зависимости соединения, т.е., если в отношении имеется многозначная зависимость, то имеется и зависимость соединения. Обратное, конечно, неверно.

Определение 6. Зависимость соединения $*(A, B, \dots, Z)$ называется **нетривиальной зависимостью соединения**, если выполняется два условия:

- Одно из множеств атрибутов A, B, \dots, Z не содержит потенциального ключа отношения R .
- Ни одно из множеств атрибутов не совпадает со всем множеством атрибутов отношения R .

Для удобства работы сформулируем это определение так же и в отрицательной форме:

Определение 7. Зависимость соединения $*(A, B, \dots, Z)$ называется **тривиальной зависимостью соединения**, если выполняется одно из условий:

- Либо все множества атрибутов A, B, \dots, Z содержат потенциальный ключ отношения R .
- Либо одно из множеств атрибутов совпадает со всем множеством атрибутов отношения R .

Определение 8. Отношение R находится в **пятой нормальной форме (5НФ)** тогда и только тогда, когда любая имеющаяся зависимость соединения является тривиальной.

Определения 5НФ может стать более понятным, если сформулировать его в отрицательной форме:

Определение 9. Отношение R не находится в **5НФ**, если в отношении найдется нетривиальная зависимость соединения.

1.4. АЛГОРИТМ ПРИВЕДЕНИЯ К 5НФ ВЫВОДЫ

Приведение к 5НФ

На предыдущем занятии был описан алгоритм нормализации как алгоритм приведения отношений к 3НФ. Теперь мы можем продолжить этот алгоритм, доведя его до алгоритма приведения к 5НФ.

Шаг 4 (Приведение к НФБК). Если имеются отношения, содержащие несколько потенциальных ключей, то необходимо проверить, имеются ли функциональные зависимости, детерминанты которых не являются потенциальными ключами. Если такие функциональные зависимости имеются, то необходимо провести дальнейшую декомпозицию отношений. Те атрибуты, которые зависят от детерминантов, не

являющихся потенциальными ключами выносятся в отдельное отношение вместе с детерминантами.

Шаг 5 (Приведение к 4НФ). Если в отношениях обнаружены нетривиальные многозначные зависимости, то необходимо провести декомпозицию для исключения таких зависимостей.

Шаг 5 (Приведение к 5НФ). Если в отношениях обнаружены нетривиальные зависимости соединения, то необходимо провести декомпозицию для исключения и таких зависимостей.

Выводы

При разработке базы данных можно выделить несколько уровней моделирования:

- Сама предметная область
- Модель предметной области
- Логическая модель данных
- Физическая модель данных
- Собственно база данных и приложения

Ключевые решения, определяющие качество будущей базы данных закладываются на этапе разработки логической модели данных. "Хорошие" модели данных должны удовлетворять определенным критериям:

- Адекватность базы данных предметной области
- Легкость разработки и сопровождения базы данных
- Скорость выполнения операций обновления данных (вставка, обновление, удаление)
- Скорость выполнения операций выборки данных

Первая нормальная форма (1НФ) - это обычное отношение. Отношение в 1НФ обладает следующими свойствами:

- В отношении нет одинаковых кортежей.
- Кортежи не упорядочены.
- Атрибуты не упорядочены.
- Все значения атрибутов атомарны.

Отношения, находящиеся в 1НФ являются "плохими" в том смысле, что они не удовлетворяют выбранным критериям - имеется большое

количество аномалий обновления, для поддержания целостности базы данных требуется разработка сложных триггеров.

Отношение R находится во **второй нормальной форме (2НФ)** тогда и только тогда, когда отношение находится в 1НФ и нет неключевых атрибутов, зависящих от части сложного ключа.

Отношения в 2НФ "лучше", чем в 1НФ, но еще недостаточно "хороши" - остается часть аномалий обновления, по-прежнему требуются триггеры, поддерживающие целостность базы данных.

Отношение R находится в **третьей нормальной форме (3НФ)** тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.

Отношения в 3НФ являются самыми "хорошими" с точки зрения выбранных нами критериев - устранены аномалии обновления, требуются только стандартные триггеры для поддержания ссылочной целостности.

Обобщением 3НФ на случай, когда отношение имеет более одного потенциального ключа, является нормальная форма Бойса-Кодда.

Отношение R находится в **нормальной форме Бойса-Кодда (НФБК)** тогда и только тогда, когда детерминанты всех функциональных зависимостей являются потенциальными ключами.

Нормализация отношений вплоть до нормальной формы Бойса-Кодда основывалась на понятии *функциональной зависимости* и теореме Хеза, гарантировавшей, что декомпозиция будет происходить без потерь информации.

Дальнейшая нормализация связана уже с обобщением понятия функциональной зависимости.

Атрибуты (множества атрибутов) Y и Z **многозначно зависят** от X , ($X \twoheadrightarrow Y|Z$), тогда и только тогда, когда из того, что в отношении R содержатся кортежи $r_1 = (x, y, z_1)$ и $r_2 = (x, y_1, z)$ следует, что в отношении R содержится также и кортеж $r_3 = (x, y, z)$.

Корректность дальнейшей декомпозиции основывается на теореме Фейджина, которая говорит о том, что декомпозиция отношения на две

проекция является декомпозицией без потерь тогда и только тогда, когда в отношении имеется некоторая многозначная зависимость.

Если в отношении имеется функциональная зависимость, то автоматически имеется и тривиальная многозначная зависимость, определяемая этой функциональной зависимостью.

Многозначная зависимость $X \twoheadrightarrow Y | Z$ называется **нетривиальной многозначной зависимостью**, если не существует функциональных зависимостей $X \rightarrow Y$ и $X \rightarrow Z$.

Отношение R находится в **четвертой нормальной форме (4НФ)** тогда и только тогда, когда отношение находится в НФБК и не содержит *нетривиальных* многозначных зависимостей.

Имеют место зависимости специального вида, когда отношение не может быть подвергнуто декомпозиции без потерь на две проекции, но может быть декомпозировано на большее число проекций. Такие зависимости называются зависимостями соединения и являются обобщением понятия многозначной зависимости.

Отношение R находится в **пятой нормальной форме (5НФ)** тогда и только тогда, когда *любая имеющаяся зависимость соединения является тривиальной*.

1.5. ДЕНОРМАЛИЗАЦИЯ

Заканчивая разговор о нормализации, необходимо упомянуть и о *денормализации*, которая определяется как намеренное приведение структуры реляционных отношений, к форме нарушающей требования НФ. В процессе нормализации реляционные отношения обычно подвергаются декомпозиции, разделяются на несколько отношений. При этом для чтения данных из нескольких отношений приходится проводить операции соединения, которые требуют дополнительного времени.

В большинстве случаев, потери времени на соединение таблиц относительно невелики, и важнее спроектировать такую структуру БД, в которой не будет аномалий модификации. Но в ряде случаев, забота о быстродействии системы вынуждает разработчиков согласиться с некоторой избыточностью хранимых данных и отсутствия у отдельных реляционных отношений старших нормальных форм. Это можно

рассматривать в качестве крайней меры, и решение о денормализации надо принимать очень осторожно.

Соберем воедино результаты анализа критериев, по которым мы хотели оценить влияние логического моделирования данных на качество физических моделей данных и производительность базы данных:

Критерий	Отношения слабо нормализованы (1НФ, 2НФ)	Отношения сильно нормализованы (3НФ)
Адекватность базы данных предметной области	ХУЖЕ (-)	ЛУЧШЕ (+)
Легкость разработки и сопровождения базы данных	СЛОЖНЕЕ (-)	ЛЕГЧЕ (+)
Скорость выполнения вставки, обновления, удаления	МЕДЛЕННЕЕ (-)	БЫСТРЕЕ (+)
Скорость выполнения выборки данных	БЫСТРЕЕ (+)	МЕДЛЕННЕЕ (-)

Как видно из таблицы, более сильно нормализованные отношения оказываются лучше спроектированы (три плюса, один минус). Они больше соответствуют предметной области, легче в разработке, для них быстрее выполняются операции модификации базы данных. Правда, это достигается ценой некоторого замедления выполнения операций выборки данных.

У слабо нормализованных отношений единственное преимущество - если к базе данных обращаться только с запросами на выборку данных, то для слабо нормализованных отношений такие запросы выполняются быстрее. Это связано с тем, что в таких отношениях уже как бы произведено соединение отношений и на это не тратится время при выборке данных.

Таким образом, выбор степени нормализации отношений зависит от характера запросов, с которыми чаще всего обращаются к базе данных.

1.6. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ (OLTP и OLAP-системы)

Можно выделить некоторые классы систем, для которых больше подходят сильно или слабо нормализованные модели данных.

Сильно нормализованные модели данных хорошо подходят для так называемых **OLTP-приложений** (*On-Line Transaction Processing (OLTP)-оперативная обработка транзакций*). Типичными примерами OLTP-приложений являются системы складского учета, системы заказов билетов, банковские системы, выполняющие операции по переводу денег, и т.п. Основная функция подобных систем заключается в выполнении большого количества коротких транзакций. Сами транзакции выглядят относительно просто, например, "снять сумму денег со счета А, добавить эту сумму на счет В". Проблема заключается в том, что, во-первых, транзакций очень много, во-вторых, выполняются они одновременно (к системе может быть подключено несколько тысяч одновременно работающих пользователей), в-третьих, при возникновении ошибки, транзакция должна целиком откатиться и вернуть систему к состоянию, которое было до начала транзакции (не должно быть ситуации, когда деньги сняты со счета А, но не поступили на счет В). Практически все запросы к базе данных в OLTP-приложениях состоят из команд вставки, обновления, удаления. Запросы на выборку в основном предназначены для предоставления пользователям возможности выбора из различных справочников. Большая часть запросов, таким образом, известна заранее еще на этапе проектирования системы. Таким образом, критическим для OLTP-приложений является скорость и надежность выполнения коротких операций обновления данных. Чем выше уровень нормализации данных в OLTP-приложении, тем оно, как правило, быстрее и надежнее. Отступления от этого правила могут происходить тогда, когда уже на этапе разработки известны некоторые часто возникающие запросы, требующие соединения отношений и от скорости выполнения которых существенно зависит работа приложений. В этом случае можно пожертвовать нормализацией для ускорения выполнения подобных запросов.

Другим типом приложений являются так называемые **OLAP-приложения** (*On-Line Analytical Processing(OLAP) - оперативная аналитическая обработка данных*). Это обобщенный термин,

характеризующий принципы построения *систем поддержки принятия решений (Decision Support System - DSS)*, *хранилищ данных (Data Warehouse)*, *систем интеллектуального анализа данных (Data Mining)*. Такие системы предназначены для нахождения зависимостей между данными (например, можно попытаться определить, как связан объем продаж товаров с характеристиками потенциальных покупателей), для проведения анализа "что если...". OLAP-приложения оперируют с большими массивами данных, уже накопленными в OLTP-приложениях, взятыми из электронных таблиц или из других источников данных. Такие системы характеризуются следующими признаками:

- Добавление в систему новых данных происходит относительно редко крупными блоками (например, раз в квартал загружаются данные по итогам квартальных продаж из OLTP-приложения).
- Данные, добавленные в систему, обычно никогда не удаляются.
- Перед загрузкой данные проходят различные процедуры "очистки", связанные с тем, что в одну систему могут поступать данные из многих источников, имеющих различные форматы представления для одних и тех же понятий, данные могут быть некорректны, ошибочны.
- Запросы к системе являются нерегламентированными и, как правило, достаточно сложными. Очень часто новый запрос формулируется аналитиком для уточнения результата, полученного в результате предыдущего запроса.
- Скорость выполнения запросов важна, но не критична.

Данные OLAP-приложений обычно представлены в виде одного или нескольких гиперкубов, измерения которого представляют собой справочные данные, а в ячейках самого гиперкуба хранятся собственно данные. Например, можно построить гиперкуб, измерениями которого являются: время (в кварталах, годах), тип товара и отделения компании, а в ячейках хранятся объемы продаж. Такой гиперкуб будет содержать данных о продажах различных типов товаров по кварталам и подразделениям. Основываясь на этих данных, можно отвечать на вопросы вроде "у какого подразделения самые лучшие объемы продаж в текущем году?", или "каковы тенденции продаж отделений Юго-Западного региона в текущем году по сравнению с предыдущим годом?"

Возвращаясь к проблеме нормализации данных, можно сказать, что в системах OLAP, использующих реляционную модель данных (ROLAP), данные целесообразно хранить в виде слабо нормализованных отношений,

содержащих заранее вычисленные основные итоговые данные. Большая избыточность и связанные с ней проблемы тут не страшны, т.к. обновление происходит только в момент загрузки новой порции данных. При этом происходит как добавление новых данных, так и пересчет итогов.