

# Раздел 3. Проектирование баз данных

## Тема 3.4.

### **Нормализация схемы реляционной БД.**

1. Основные понятия нормализации.

2. Последовательность действий, выполняемая в процессе нормализации.

Классическая технология проектирования реляционных баз данных с использованием декомпозиции связана с теорией нормализации, основанной на анализе функциональных зависимостей между атрибутами отношений.

Функциональные зависимости определяют устойчивые отношения (связи) между объектами и их свойствами в рассматриваемой предметной области.

Для чего нужна нормализация схемы БД?

Рассмотрим для примера конкретную схему отношений и проанализируем её недостатки.

Предположим, что **данные о студентах, факультетах, специальностях**, включены в таблицу со следующей схемой отношения:

**СТУДЕНТ** (Код студента, Фамилия, Название факультета, Название специальности).

Код студента	Фамилия	Название факультета	Название специальности
025001	Иванов	Экономики	Менеджер-экономист
025002	Петров	ФПМИ	Математик-программист
025101	Сидоров	Журналистики	репортер

## Отношение «Студент»

Код студента	Фамилия	Название факультета	Название специальности
025001	Иванов	Экономики	Менеджер-экономист
025002	Петров	ФПМИ	Математик-программист

Эта схема отношений обуславливает **следующие недостатки** соответствующей реальной базы данных:

- **Дублирование информации (избыточность)**. У студентов, обучающихся на одном факультете, будет повторяться название факультета. Для разных факультетов могут повторяться специальности.
- **Потенциальная противоречивость (аномалии обновления)**. Если, например, изменится название специальности, то изменяя её в одном кортеже (у одного студента), необходимо изменять и во всех других кортежах, где она присутствует.
- Потенциальная **возможность потери сведений (аномалии удаления)**. При удалении информации о всех студентах, поступающих на определенную специальность, мы теряем все сведения об этой специальности.
- Потенциальная **возможность невключения информации в базу данных (аномалии включения)**. В базе данных будут отсутствовать сведения о специальности, если на ней нет обучающихся студентов.

**Построение рационального варианта схем отношений**  
(обладающего лучшими свойствами при операциях включения, модификации и удаления данных, чем все остальные наборы схем) и осуществляется путем так называемой **нормализации схем отношений**.

Нормализация производится в **несколько этапов**, при этом **каждая последующая итерация** соответствует **нормальной форме более высокого уровня** и обладает **лучшими свойствами** по сравнению с предыдущей.

**Каждой нормальной форме** соответствует **некоторый определенный набор ограничений**, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

Таким образом - Один и тот же набор данных в реляционной модели можно **представить различными способами**.

Процесс **нормализации** данных позволяет решить вопрос о наиболее эффективной их структуре, обладающей **минимальной избыточностью**.

Это позволяет решить следующие **задачи**:

- a)** исключить ненужное повторение данных,
- b)** обеспечить быстрый доступ к данным,
- c)** обеспечить целостность данных.

Обычно говорят только об уменьшении избыточности данных и поэтому часто **нормализацией называют процесс уменьшения избыточности данных** в реляционной БД. Это не совсем верно.

В теории реляционных БД обычно выделяется **следующая последовательность нормальных форм**:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);
- нормальная форма Бойса— Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или форма проекции-соединения (5NF или PJNF).

**На начальном этапе** схема отношений **должна находиться в первой нормальной форме (1НФ)**.

**Без этого эта схема не будет считаться схемой РБД.**

При некоторых предположениях о данных **третья нормальная форма является наилучшим искомым вариантом**.

Если эти предположения не выполняются, то процесс нормализации продолжается и отношение преобразуется к следующей нормальной форме.

## Основные свойства нормальных форм:

- ✓ каждая следующая нормальная форма **уменьшает избыточность** данных и улучшает свойства предыдущей;
- ✓ при переходе к следующей нормальной форме **свойства предыдущих нормальных форм сохраняются**.

Перевод отношения в следующую нормальную форму осуществляется **методом «декомпозиции без потерь»**, т. е. разбиением исходной таблицы на несколько связанных.

Такая декомпозиция должна обеспечить **равенство результатов выборки** (поисковых запросов)

- из **исходного** отношения и
- выборки, основанных на **совокупности** полученных в результате нормализации отношений (таблиц).



Итак, в основе классического процесса проектирования лежит последовательность переходов от предыдущей нормальной формы к последующей (декомпозиции).

Однако в процессе декомпозиции мы сталкиваемся **с проблемой обратимости**, то есть возможности **восстановления исходной схемы**.

Таким образом, **декомпозиция должна сохранять эквивалентность схем БД** при замене одной схемы на другую.

Схемы БД **называются эквивалентными**, если содержание исходной БД может быть получено путем **естественного соединения отношений**, входящих в результирующую схему, и при этом **не появляется новых кортежей в исходной БД**.

При выполнении эквивалентных преобразований **сохраняется множество исходных фундаментальных функциональных зависимостей** между атрибутами отношений.

**Функциональной зависимостью** набора атрибутов **B** отношения **R** от набора атрибутов **A** того же отношения, обозначаемой как  **$R.A \rightarrow R.B$  или  $A \rightarrow B$**

**называется такое соотношение проекций  $R[A]$  и  $R[B]$ , при котором в каждый момент времени любому элементу проекции  $R[A]$  соответствует только один элемент проекции  $R[B]$ , входящий вместе с ним в какой-либо кортеж отношения  $R$ .**

Студ_ном	Студ_Фам	Студ_предм	Студ_оценка
27711	Иванов	БД	5
27601	Петров	БД	8

**$R.Студ\_ном \rightarrow R.Студ\_Фам$  или  $Студ\_ном \rightarrow Студ\_Предм$**

Если в отношении существует несколько функциональных зависимостей, то **каждый атрибут или набор атрибутов, от которого зависит другой атрибут, называется детерминантом отношения.**

Студ_ном	Студ_Фам	Студ_билет	Студ_оценка
27711	Иванов	5778	5
27601	Петров	3562	8

**Студ\_ном** -> Студ\_Фам, **Студ\_ном** -> Студ\_билет

Атрибут **Студ\_ном** – детерминант данного отношения

**Функциональные зависимости определяют не текущее состояние БД, а **все возможные ее состояния**, то есть они отражают те связи между атрибутами, которые присущи реальному объекту, моделируемому с помощью схемы БД.**

**Набор функциональных зависимостей задает разработчик, системный аналитик, исходя из глубокого системного анализа предметной области.**

**Рекомендации**, которых следует придерживаться при нормализации таблиц.

1. Разрабатывайте схему данных таким образом, чтобы можно было бы легко объяснить ее, т. е. не комбинируйте атрибуты независимых объектов и не создавайте сложные связи.
2. Разрабатывайте схему данных таким образом, чтобы исключить возможность появления аномалий обновления.
3. Разрабатывайте схему данных таким образом, чтобы в связях участвовали только первичные (можно допустить потенциальные) и внешние ключи. Это позволит избежать появления подложных записей.
4. **Не обязательно нормализовывать все таблицы** БД, и притом не обязательно до форм высших порядков.

# Раздел 3. Проектирование баз данных

## Тема 3.4.

### **Нормализация схемы реляционной БД.**

1. Основные понятия нормализации.
2. Последовательность действий, выполняемая в процессе нормализации.

**Первая нормальная форма** требует соответствия исходной таблицы общим требованиям, **предъявляемым к реляционным отношениям**, т. е.:

- 1) таблица не должна иметь **повторяющихся** записей;
- 2) все атрибуты должны быть **простыми** (скалярными).

**Первое требование** легко решается **введением ключевого поля**, **однозначно** определяющего остальные поля таблицы.

**Второе требование** постулирует, чтобы **в каждой ячейке было** представлено только **одно значение**, а не массив или перечисление.

Составные поля должны быть разложены на простые, а **многозначные – вынесены в отдельные таблицы** или разложены по отдельным строкам (с дублированием значений всех остальных полей).

Т.о. - отношение находится **в первой нормальной форме** тогда и только тогда, когда на пересечении каждого столбца и каждой строки **находятся только элементарные значения атрибутов.**

Отношения, находящиеся в первой нормальной форме, часто называют **просто нормализованными** отношениями.

Соответственно, **ненормализованные** отношения могут интерпретироваться как **таблицы с неравномерным заполнением**, например таблица «Расписание», которая представлена на следующем слайде:



Преподаватель	День	Пара	Дисц	Группа
Иванов	ПНД	1	БД	111101
	ВТР	1	БД	111102
	ВТР	3	АСУ	91101
Петров	ПНД	1	ВМ	111101
	ВТР	1	ВМ	111101
Киров	ПНД	2	ИТ	111101

Преподаватель	День	Пара	Дисц	Группа
Иванов	ПНД	1	БД	111101
Иванов	ВТР	1	БД	111102
Иванов	ВТР	3	АСУ	91101
Петров	ПНД	1	ВМ	111101
Петров	ВТР	1	ВМ	111101
Киров	ПНД	2	ИТ	111101

Таблица находится **во второй нормальной форме**, если:

- 1) она удовлетворяет условиям первой нормальной формы;
- 2) любое **поле, не входящее в ключ**, должно однозначно и полно идентифицироваться значением первичного ключа.

Если первичный ключ является составным, то остальные поля должны **зависеть от его полного выражения**, а не от части (**неключевые атрибуты** должны характеризоваться **полной функциональной зависимостью** от первичного ключа).

Функциональная зависимость  $R.A \rightarrow R.B$  называется **полной**, если набор атрибутов  $B$  функционально зависит от  $A$  и не зависит функционально от любого подмножества  $A$ ,

то есть  $R.A \rightarrow R.B$  называется полной, если:

любое  $A1$  из  $A \Rightarrow R.A1 \not\rightarrow R.B$ , что читается следующим образом:

для любого  $A1$ , являющегося подмножеством  $A$  отношения  $R$ ,  $R.B$  функционально не зависит от  $R.A1$ , в противном случае зависимость  $R.A \rightarrow R.B$  называется **неполной**.

Т.о. - отношение находится **во второй нормальной форме** тогда и только тогда, когда оно находится **в первой нормальной форме** и **не содержит неполных функциональных зависимостей** непервичных (неключевых) атрибутов **от отдельных атрибутов составного первичного ключа**.

ФИО	Номер_Зач_КН	Группа	Дисципл	Оценка
Иванов	45900	111101	БД	9

**Первичным ключом** отношения может быть (Номер, зач.кн.. Дисциплина), однозначно определяющие строки данного отношения.

Но **атрибуты ФИО и Группа** зависят только от части первичного ключа — от значения атрибута **Номер зач, кн.**, поэтому мы должны констатировать **наличие неполных функциональных зависимостей** в данном отношении.

Для приведения данного отношения ко **второй** нормальной форме следует **разбить его на проекции**, при этом должно быть соблюдено условие **восстановления исходного отношения без потерь**.

Таковыми проекциями могут быть два отношения:

(ФИО, Номер.зач.кн., Группа) и

(Номер зач.кн., Дисциплина. Оценка)

Этот набор отношений **не содержит неполных функциональных зависимостей**, и поэтому эти два отношения находятся **во второй** нормальной форме.

Таблица находится **в третьей нормальной форме**, если:

- 1) она удовлетворяет условиям второй нормальной формы;
- 2) ни одно из полей таблицы, не входящих в ключ, **не должно идентифицироваться** с помощью другого поля, не входящего в ключ

(иначе, такое отношение **не должно иметь неключевых атрибутов**, которые бы находились **в транзитивной зависимости от первичного ключа**).

Функциональная зависимость  $R.A \rightarrow R.B$  называется *транзитивной*, если существует набор атрибутов  $C$  такой, что:

1.  $C$  не является подмножеством  $A$ .
2.  $C$  не включает в себя  $B$  ( $B$  не является подмножеством  $C$ ).
3. Существует функциональная зависимость  $R.A \rightarrow R.C$ .
4. Не существует функциональной зависимости  $R.C \rightarrow R.A$ .
5. Существует функциональная зависимость  $R.C \rightarrow R.B$ .

Другими словами, существует зависимость

$$R.A \rightarrow R.C \rightarrow R.B$$

Итак, отношение находится **в третьей нормальной форме** тогда и только тогда, когда оно находится **во второй нормальной форме** и **не содержит транзитивных зависимостей**.

На примере следующего отношения видим, что оно **не находится в третьей НФ**:

ФИО	Ном_Зач_Кн	Группа	Факультет	Спец-ть	Вып_Каф
Иванов	2345	111101	ИИФ	ИСиТ	АИС

В этом **случае у нас есть следующие функциональные зависимости**:

Номер зач .кн. -> ФИО

Номер зач.кн. -> **Группа**

Номер зач.кн. -> Факультет

Номер зач.кн. -> Специальность

Номер зач.кн. -> **Выпускающая кафедра**

**Но при этом еще есть зависимости:**

**Группа** -> Факультет

**Группа** -> Специальность

**Группа** -> **Выпускающая кафедра**

**и еще:** **Выпускающая кафедра** -> Факультет

**Получается эти зависимости образуют транзитивные группы.**



Для того чтобы избежать этого, мы можем предложить следующий набор отношений:

(**Номер. зач. кн.**, ФИО. Специальность. Группа)

(**Группа.** , Выпускающая кафедра)

(**Выпускающая кафедра**, Факультет)

*Первичные ключи отношений выделены.*

Полученный набор отношений находится **в третьей** нормальной форме. При этом **в случае естественного соединения таблиц** мы:

- ✓ не потеряем ни одной строки и
- ✓ не получим лишних кортежей данных.

**На этом процесс нормализации обычно заканчивается.**

Если же необходима **более детальная проработка данных**, то дополнительно рассматривают требования нормальных форм более высокого порядка.

**Третья** нормальная форма постулирует **отсутствие зависимости одних неключевых атрибутов от других неключевых атрибутов**, забывая о том, что в ряде случаев **может наблюдаться зависимость части ключевого составного атрибута от некоторого неключевого атрибута**.

Если в таблице имеется такая зависимость, то необходимо перейти **к усиленной третьей нормальной форме**, или **форме Бойса – Кодда (BCNF)**.

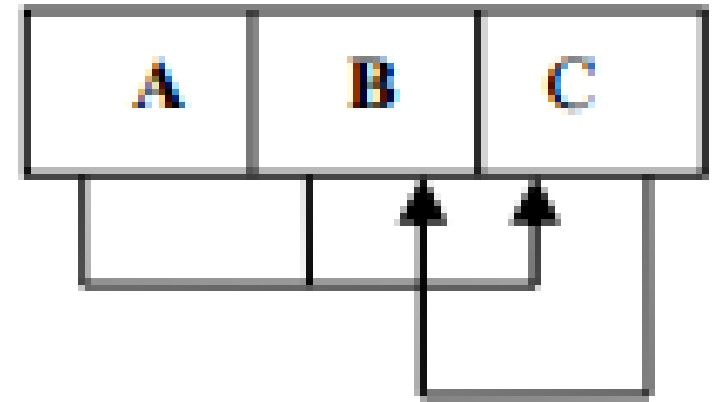
Таблица находится **в нормальной форме Бойса – Кодда**, если:

- 1) удовлетворяет требованиям третьей нормальной формы;
- 2) в ней **отсутствуют зависимости атрибутов составного ключа от остальных неключевых атрибутов**.

Например:

Неключевой **атрибут С** функционально **зависит** от составного ключа (**A + B**).

**Атрибут В** в свою очередь функционально **зависит от атрибута С**, т.е. **неключевой атрибут С** выступает в роли **детерминанта** для атрибута В.



Не путать с **неполной** функциональной зависимостью – там наоборот неключевой атрибут **С** зависел бы от атрибута **В** – части составного ключа A,B)

Такая взаимозависимость **требует дальнейшей декомпозиции таблицы**.

Т.о., если в отношении существует несколько функциональных зависимостей, то **каждый атрибут или набор атрибутов, от которого зависит другой атрибут,** называется **детерминантом отношения**.

Студ_ном	Студ_Фам	Студ_билет	Студ_оценка
27711	Иванов	5778	5
27601	Петров	3562	8

**Студ\_ном** -> Студ\_Фам,  
**Студ\_ном** -> Студ\_билет

Атрибут **Студ\_ном** – детерминант данного отношения

Однако в данном отношении можно выделить еще один **потенциальный ключ** отношения: **Студ\_билет**

НФ БК **учитывает все потенциальные ключи**, которые входят в отношения.

Если отношение имеет **единственный потенциальный ключ**, то 3-я НФ и НФ БК – эквивалентны.

**Потенциальным (возможным) ключом** отношения называется набор атрибутов отношения, который **полностью и однозначно (функционально полно) определяет значения всех остальных атрибутов отношения**, то есть возможный ключ — это набор атрибутов, также однозначно определяющий кортеж отношения.

<b>Студ_ном</b>	Студ_Фам	<b>Студ_билет</b>	Студ_оценка
27711	Иванов	5778	5
27601	Петров	3562	8

Отношение находится **в нормальной форме Бойса—Кодда**, если оно **находится в третьей нормальной форме** и **каждый детерминант** отношения является **ВОЗМОЖНЫМ КЛЮЧОМ** отношения.

*Рассмотрим отношение, моделирующее сдачу студентом текущих экзаменов.*

Ном_Зач_КН	Идентификатор_Студ	Дисц-на	Дата	Оценка
Б-2345	2345	БД	14.05.13	9

**Возможными (составными)** ключами отношения являются :

**Нонер\_зач.кн, Дисциплина, Дата** и

**Идентификатор\_студента, Дисциплина, Дата.**

(только они **однозначно определяют кортежи** этого отношения)

Какие функциональные зависимости у нас имеются?

## Функциональные зависимости:

1. Номер\_зач.кн, Дисциплина. Дата -> Оценка;
2. Идентификатор\_студента, Дисциплина. Дата -> Оценка;
3. Номер зач.кн. -> Идентификатор\_студента;
4. Идентификатор\_студента -> Номер зач.кн.

Но **зависимости 3 и 4 избыточные**, так как это **детерминанты**, которые не являются возможными ключами отношений в данном отношении.

Действительно, если оставить в данном отношении ключом только **Номер зач.кн.**, мы **не сможем однозначно выделить кортеж** по отдельному предмету с конкретной оценкой.

То же самое и при использовании ключа **Идентификатор\_студента**

Для приведения отношения к нормальной форме Бойса—Кодда надо **разделить исходное отношение**, например, **на два** со следующими схемами:

(**Идентификатор\_студента**. Дисциплина. Дата. Оценка)

(**Идентификатор\_студента**. Номер зач.кн.)

или наоборот:

(**Номер зач.кн.**, Дисциплина. Дата, Оценка)

(**Номер зач.кн.**, Идентификатор\_студента)

Эти схемы равнозначны с точки зрения теории нормализации, поэтому выбирать проектировщикам следует исходя из некоторых дополнительных рассуждений.



# Нормальные формы высших порядков

## 4NF и 5NF (PJ/NF)

Четвертая нормальная форма устраняет **нетривиальные многозначные** зависимости.

Отношение находится в 4НФ, если оно находится в **BCNF** и в нем **отсутствуют нетривиальные** многозначные зависимости.

*Определение:*

В отношении  $R(A, B, C)$  существует **многозначная зависимость** (multi valid dependence, **MVD**)  $R.A \twoheadrightarrow R.B$  в том и только в том случае, если **множество значений  $B$** , соответствующее паре значений  $A$  и  $C$ , **зависит только от  $A$  и не зависит от  $C$** .

Различают **тривиальные** и **нетривиальные** многозначные зависимости.

## Нормальные формы высших порядков **4NF и 5NF (PJ/NF)**

Для случая трех атрибутов  $X, Y, Z$  **многозначная зависимость**

$X \twoheadrightarrow Y|Z$  будет **тривиальной**, если не существует функциональных зависимостей  $X \rightarrow Y$  и  $X \rightarrow Z$ .

Другими словами, **многозначная зависимость** является **нетривиальной**, если наблюдаемые многозначные зависимости  $X \twoheadrightarrow Y$  и  $X \twoheadrightarrow Z$  фактически являются функциональными **зависимостями от первичного ключа отношения**.

Тривиальная многозначная зависимость не нарушает 4НФ, а **нетривиальная требует дальнейшей декомпозиции отношения**, причем чаще всего сразу на три отношения.

Последней нормальной формой является **пятая нормальная форма 5NF**, которая связана с анализом нового вида зависимостей, зависимостей **«проекции соединения»** (*project-join зависимости, обозначаемые как PJ-зависимости*).

Этот вид зависимостей является в некотором роде обобщением многозначных зависимостей.

Отношение  $R(X, Y, \dots, Z)$  удовлетворяет **зависимости соединения**  $(X, Y, \dots, Z)$  в том и только в том случае, когда  **$R$  восстанавливается без потерь путем соединения своих проекций на  $X, Y, \dots, Z$** .  
Здесь  $X, Y, \dots, Z$  — наборы атрибутов отношения  $R$ .

Отношение  $R$  находится в пятой нормальной форме (нормальной форме проекции-соединения — PJ/NF) в том и только в том случае, когда **любая зависимость соединения в  $R$  следует из существования некоторого возможного ключа в  $R$ .**

Пятая нормальная форма **редко используется на практике**. В большей степени она является теоретическим исследованием.

Спасибо за внимание!