

Севастопольский государственный университет  
Кафедра «Информационные системы»

# Управление данными

курс лекций

лектор:  
ст. преподаватель кафедры ИС Абрамович А.Ю.



## Лекция 5

**Жизненный цикл БД.**

**Нормализация. Денормализация**

# ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Полный цикл разработки базы данных включает **концептуальное, логическое и физическое проектирование**.

**Логическое проектирование базы данных (БД)** – это процесс создания структуры и организации данных в БД с учетом требований и целей предприятия или организации. Включает в себя **определение сущностей (таблиц), атрибутов (столбцов) и связей между ними**, а также **определение правил и ограничений** для хранения и обработки данных.

Цель второй фазы проектирования базы данных состоит в создании **эффективной и гибкой структуры данных, которая позволит эффективно хранить, обрабатывать и извлекать информацию**. Структура должна удовлетворять требованиям надежности, безопасности и целостности данных.

На этом этапе уже должно быть известно, какая СУБД будет использоваться в качестве целевой - реляционная, сетевая, иерархическая или объектно-ориентированная, но игнорируются все остальные характеристики выбранной СУБД, например, любые особенности физической организации ее структур хранения данных и построения индексов.

# ПОДХОДЫ И МЕТОДЫ К ЛОГИЧЕСКОМУ ПРОЕКТИРОВАНИЮ БД

## TOP-DOWN ПОДХОД

Top-down подход к логическому проектированию БД начинается с **общего представления о системе и постепенно уточняется до более детального уровня**. В этом подходе разработчики начинают с определения общих сущностей и связей между ними, а затем постепенно добавляют дополнительные детали и атрибуты. Top-down подход позволяет разработчикам иметь общее представление о системе и ее структуре, что облегчает понимание и поддержку БД в будущем.

## BOTTOM-UP ПОДХОД

Bottom-up подход к логическому проектированию БД начинается с **определения конкретных атрибутов и связей между ними, а затем постепенно объединяет их в более общие сущности**. В этом подходе разработчики начинают с низкого уровня детализации и постепенно строят более общую структуру БД. **Bottom-up подход позволяет разработчикам сосредоточиться на конкретных деталях и атрибутах**, что может быть полезно при работе с большими и сложными БД.

## ER-МОДЕЛИРОВАНИЕ

ER-моделирование (Entity-Relationship Modeling) – это **метод моделирования, который используется для описания сущностей, атрибутов и связей между ними в БД**. ER-моделирование позволяет разработчикам визуализировать структуру БД и легко понять связи между сущностями.

## НОРМАЛИЗАЦИЯ БД

Нормализация БД – это **процесс разделения таблиц на более мелкие и более связанные таблицы для устранения избыточности данных и обеспечения целостности БД**. Нормализация помогает улучшить эффективность и гибкость БД, а также уменьшить объем хранимых данных.

## ДЕНОРМАЛИЗАЦИЯ БД

Денормализация БД – это **процесс объединения таблиц в одну или несколько таблиц для увеличения производительности и упрощения запросов**. Денормализация может быть полезна в случаях, когда требуется быстрый доступ к данным или когда объем данных невелик.

# НОРМАЛИЗАЦИЯ

**Нормализация базы данных** - это метод создания таблиц БД путем разделения (или декомпозиции) таблицы большего размера на небольшие логические единицы. В данном методе учитываются требования, предъявляемые к среде БД.

**Нормализация базы данных (БД)** - это метод проектирования реляционных БД, который **помогает правильно структурировать таблицы данных**. Процесс направлен на **создание системы с четким представлением информации и взаимосвязей, без избыточности и потери данных**.

## Цель нормализации

**Идентифицировать подходящий набор отношений, который будет адекватно поддерживать требования к данным некоторой предметной области:**

- минимальное количество атрибутов;
- атрибуты, которые логически тесно связаны, должны находиться в одном отношении;
- минимальная избыточность, т. е. каждый атрибут должен быть представлен только один раз, за исключением атрибутов внешних ключей.

# Зачем нужна нормализация базы данных?

Нормализация - это итеративный процесс. Каждый последующий шаг разбивает таблицу на более легкую в управлении информацию, чем повышается общая логичность системы и простота работы с ней.

Нормализация **позволяет разработчику БД оптимально распределять атрибуты по таблицам**. Данная методика **избавляет от:**

- атрибутов с несколькими значениями;
- задвоения или повторяющихся атрибутов;
- атрибутов, не поддающихся классификации;
- атрибутов с избыточной информацией;
- атрибутов, созданных из других признаков.

**Необязательно** выполнять *полную нормализацию БД*. Но она гарантирует полноценно функционирующую информационную среду. Этот метод:

- позволяет создать структуру базы данных, подходящую для общих запросов;
- сводит к минимуму избыточность данных, что повышает эффективность использования памяти на сервере БД;
- гарантирует максимальную целостность данных, устраняя аномалий вставки, обновления и удаления.

## Избыточность баз данных и аномалии

Избыточность подразумевает **наличие нескольких копий одних и тех же данных в базе данных**. Эта проблема возникает, когда база данных не нормализована.

Student_ID	Имя	Контакты	Университет	Институт
100	Петров Пётр	897654335	СевГУ	ИИТ
101	Иванов Иван	987544679	СевГУ	ГПИ
102	Сидоров Семён	987554356	СевГУ	ИИТ
103	Сидорова Симона	987643246	СевГУ	ИИТ

**Проблемы, вызванные избыточностью, следующие:**

- аномалия вставки;
- аномалия удаления;
- аномалия обновления.

**Для устранения подобных аномалий используется нормализация базы данных.**

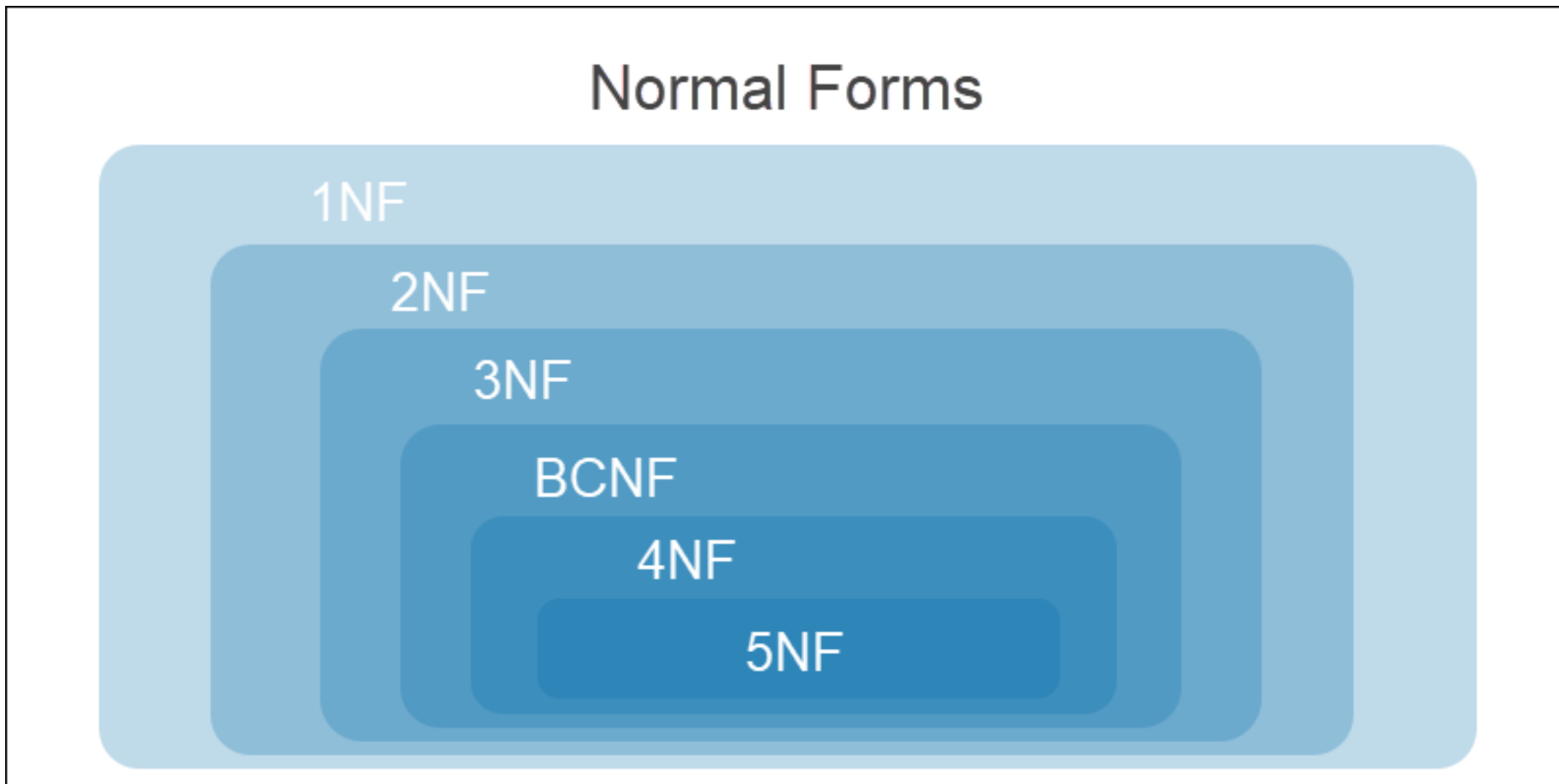
Книга	Студент
Книга 1	Иванов
Книга 2	Петров
Книга 3	Иванов

Данные  
**дублированные,**  
но **не избыточные**

Книга	Студент	Телефон
Книга 1	Иванов	9631440779
Книга 2	Петров	9170550011
Книга 3	Иванов	9631440779

Данные  
**избыточные**

Все уровни нормализации считаются **накопительными**. **Прежде чем перейти к следующему этапу, выполняются все требования к текущей форме.**



**+ Доменно-ключевая нормальная форма, Шестая нормальная форма**



# Первая нормальная форма (1НФ)

Схема отношения **R** находится в **1НФ**, если для любого атрибута **A** из схемы **R** значения являются атомарными.

*Каждый атрибут отношения должен хранить одно-единственное значение и не являться ни списком, ни множеством значений.*

**Приведение отношения к 1НФ – довольно простая операция.** Необходимо просмотреть схему отношения и разделить составные атрибуты на различные строки/столбцы. Возможно, эту операцию придется повторить несколько раз до тех пор, пока каждый из атрибутов не станет атомарным.

Отношение, находящееся в 1НФ, **должно удовлетворять следующим свойствам:**

- в отношении **нет одинаковых кортежей**;
- кортежи **не упорядочены**;
- атрибуты **не упорядочены и различаются** по наименованию;
- все значения **атрибутов атомарны** (когда на пересечении любого столбца и любой записи находится атомарное значение).

Наименование	Город	Адрес	Вид	Контактные лица
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И., зам. дир., тел (3254)76-15-95
				Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	Васильев В.В., директор, тел (3254)74-57-45

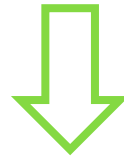
Очевидно, что **атрибут «контактные лица» не является атомарным**, поскольку в нем попадают списки из нескольких лиц.



Наименование	Город	Адрес	Вид	Контактные лица
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И., зам. дир., тел (3254)76-15-95
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	Васильев В.В., директор, тел (3254)74-57-45

Наименование	Город	Адрес	Вид	Контактные лица
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И., зам. дир., тел (3254)76-15-95
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	Васильев В.В., директор, тел (3254)74-57-45

Атрибут **«контактные лица»** не может быть назван атомарным, поскольку содержит разнородные данные, хотя и об одном лице.



Наименование	Город	Адрес	Вид	Должность	Ф.И.О.	Код города	Телефон
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	зам. дир.	Иванов И.И.	3254	76-15-95
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	нач. отд. сбыта	Петров П.П.	3254	76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	директор	Сидоров С.С.	7634	66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	директор	Васильев В.В.	3254	74-57-45

## Вторая нормальная форма (2НФ)

Схема отношения **R** находится во **2НФ** относительно множества функциональных зависимостей **F**, если она находится в **1НФ** и каждый неключевой атрибут полностью зависит от каждого ключа для **R**.

*Отношение находится во 2НФ, если оно находится в 1НФ, и при этом все неключевые атрибуты зависят только от ключа целиком, а не от какой-то его части.*

Атрибут **B** отношения **функционально зависит от атрибута A** того же отношения (атрибуты могут быть составными) в том и только в том случае, **когда в любой заданный момент времени для каждого из различных значений атрибута A обязательно существует только одно из различных значений атрибута B**.

Атрибут **B** находится в **полной функциональной зависимости от составного атрибута A**, если он **функционально зависит от A** и **не зависит функционально от любого подмножества атрибута A**.



Диаграмма функциональной зависимости (или схема ФЗ)

При постановке задачи заказчик сообщил, что в пределах каждого города наименование предприятия является уникальным, но в разных городах названия могут совпадать. Таким образом, предприятие характеризуется **составным ключом «наименование + город»**.

Очевидно, что телефонный код города зависит исключительно от самого города и никак не связан с названием предприятия. Отсюда и один из источников избыточных данных. Чтобы устранить эту избыточность, придется разбить отношение на несколько.

<u>Наименование</u>	<u>Город</u>	Адрес	Вид	Должность	Ф.И.О.	Телефон
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	зам. дир.	Иванов И.И.	76-15-95
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	нач. отд. сбыта	Петров П.П.	76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	директор	Сидоров С.С.	66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	директор	Васильев В.В.	74-57-45

<u>Город</u>	<u>Код города</u>
Владимир	3254
Курск	7634

**ОТНОШЕНИЕ С ПРОСТЫМ, ИЛИ АТОМАРНЫМ, КЛЮЧОМ, ПРИВЕДЕННОЕ К 1НФ, НАХОДИТСЯ ВО 2НФ ПО ОПРЕДЕЛЕНИЮ И В ДАННОМ ЭТАПЕ НОРМАЛИЗАЦИИ НЕ НУЖДАЕТСЯ.**

## Третья нормальная форма (3НФ)

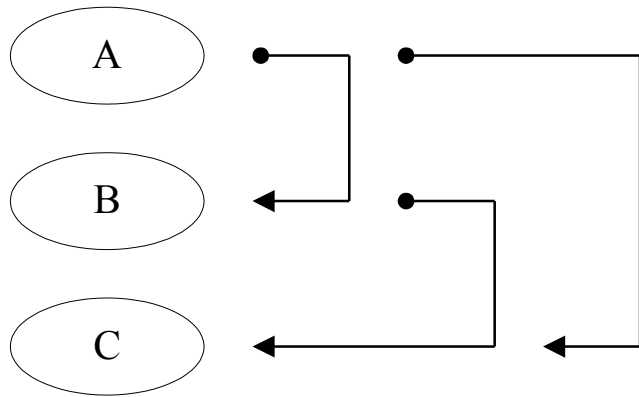
Схема отношения **R** находится в **3НФ** относительно множества функциональных зависимостей **F**, если она находится в **2НФ** и ни один из **непервичных атрибутов** в **R** не является транзитивно зависимым от ключа для **R**.

*Чтобы привести отношение к 3НФ, необходимо устранить функциональные зависимости между неключевыми атрибутами отношения. То есть, факты, хранимые в таблице, должны зависеть только от ключа.*

Атрибут **B** отношения **транзитивно функционально** зависит от атрибута **A** того же отношения (атрибуты могут быть составными) в том и только в том случае, **если** существует такой атрибут **C**, что имеются функциональные зависимости между атрибутами **A** и **C**, а также между **B** и **C**.

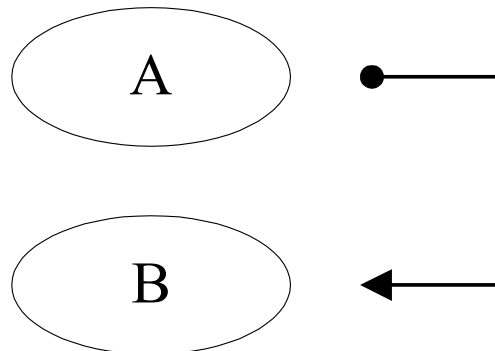
**Теорема Хита** дает возможность выполнить декомпозицию отношений без потерь информации:

- исходное отношение с ФЗ преобразуется в другие отношения, в каждом из которых атрибуты минимально зависят от первичного ключа;
- атрибут С минимально зависит от атрибута А, если выполняется минимальная слева ФЗ  $B \rightarrow C$ .

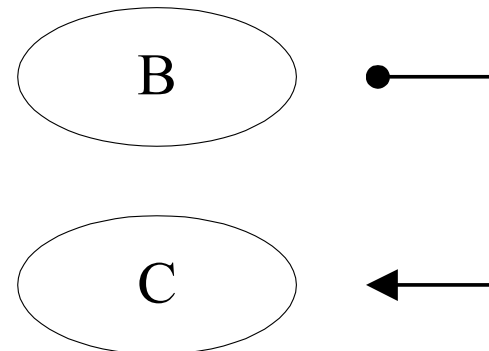


Если  **$A \rightarrow B$  и  $B \rightarrow C$ , но  $B \nrightarrow A$**  (В не является ключом), то  $A \rightarrow C$ . В этом случае С транзитивно зависит от А. Преобразование в 3НФ состоит в декомпозиции исходного отношения на два отношения.

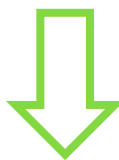
**R1**



**R2**



<u>Наименование</u>	<u>Город</u>	Адрес	Вид	Должность	Ф.И.О.	Телефон
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	зам. дир.	Иванов И.И.	76-15-95
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик	нач. отд. сбыта	Петров П.П.	76-15-35
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент	директор	Сидоров С.С.	66-65-38
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент	директор	Васильев В.В.	74-57-45



<u>Наименование</u>	<u>Город</u>	Адрес	Вид
Поршневой з-д	Владимир	ул. 2-я Кольцевая, 17	Поставщик
ООО Вымпел	Курск	ул. Гоголя, 25	Клиент
ИЧП Альфа	Владимир	ул. Пушкинская, 37	Клиент

<u>Ф.И.О.</u>	Наименование	Город	Должность	Телефон
Иванов И.И.	Поршневой з-д	Владимир	зам. дир.	76-15-95
Петров П.П.	Поршневой з-д	Владимир	нач. отд. сбыта	76-15-35
Сидоров С.С.	ООО Вымпел	Курск	директор	66-65-38
Васильев В.В.	ИЧП Альфа	Владимир	директор	74-57-45



## Нормальная форма Бойса-Кодда (НФБК)

Нормальная форма Бойса-Кодда считается уточнением ЗНФ. Она учитывает все потенциальные ключи, которые входят в отношения. Если отношение имеет **единственный потенциальный ключ**, то ЗНФ и НФБК – эквивалентны. Считается, что отношение, находящееся в НФБК, если каждый его детерминант является **потенциальным ключом**. Чтобы убедиться, что отношение находится в НФБК необходимо отыскать все его детерминанты и убедиться, что они являются потенциальными ключами.

***Детерминантом** является один атрибут или группа атрибутов, от которой полностью функционально зависит другой атрибут.*

**Нарушение требований НФБК** происходит:

- если имеются **два или более составных** ключа;
- если **перекрывается потенциальный ключ**, т.е. если какой-то атрибут входит в несколько ключей.

***Для отношений, имеющих один потенциальный ключ (первичный), НФБК является ЗНФ.***

**Пример БКНФ.** Пусть есть отношение вида:

Служащие-Проекты(Н\_служащего, Имя\_служащего, Н\_проекта, Работа\_Служащего).

**Возможными ключами отношения являются следующие пары атрибутов:**

Н\_служащего, Н\_проекта;

Имя\_служащего, Н\_проекта.

Предполагается, что Имя\_служащего так же уникально, как и Н\_служащего. Видно, что отношение Служащие-Проекты находится в 3НФ.

**В отношении есть следующие функциональные зависимости:**

Н\_служащего  $\rightarrow$  Имя\_служащего;

Н\_служащего  $\rightarrow$  Н\_проекта;

Имя\_служащего  $\rightarrow$  Н\_служащего;

Имя\_служащего  $\rightarrow$  Н\_проекта;

Н\_служащего, Н\_проекта  $\rightarrow$  Работа\_Служащего;

Имя\_служащего, Н\_проекта  $\rightarrow$  Работа\_Служащего.

**Тот факт, что имеются функциональные зависимости атрибутов отношения от атрибута, являющегося частью первичного ключа, приводит к аномалиям.**

Если вынести связь Н\_служащего→Имя\_Служащего в отдельное отношение, то получится два отношения:

1.Служащие(Н\_Служащего, Имя\_служащего).

**Возможные ключи:**

Н\_Служащего,  
Имя\_служащего.

**Зависимости:**

Н\_Служащего→Имя\_служащего;  
Имя\_служащего→Н\_Служащего.

2.Служащие\_Проекты(Н\_служащего, Н\_проекта, Работа\_Служащего).

**Возможный ключ:** Н\_служащего, Н\_проекта.

**Зависимости:** (Н\_служащего, Н\_проекта)→Работа\_Служащего.

**Такая схема находится в БКНФ.**

## Четвертая нормальная форма (4НФ)

Отношение находится в **4НФ**, если оно удовлетворяет **НФБК** и не содержит **многозначных нетривиальных зависимостей**.

В ходе проектирования БД выявлен один тип зависимости – **многозначная зависимость**.

В данном отношении имеются многозначные зависимости типа «один ко многим» (1:N).

N_отдела	ФИО_сотрудника	ФИО_клиента
011	Кот	Чижик
012	Пёс	Лебедев
011	Кот	Гусев
012	Пёс	Тупик

Если для каждого атрибута А имеется набор атрибутов В и С. Хотя атрибут В и С не зависят друг от друга. **Многозначная зависимость  $A \rightarrow B$  и  $A \rightarrow C$** .

Многозначная зависимость подразделяется на **тривиальную и не тривиальную зависимости**.

Многозначная зависимость А и В, определенных на некотором отношении R, называется **тривиальной**, если атрибут В является подмножеством атрибута А. В противном случае тривиальная зависимость является **не тривиальной**.

**Приведение базы данных к 4НФ сокращает дублирование данных, но появление новых отношений усложняет схему базы данных.**

## Пятая нормальная форма (5НФ)

**Зависимость соединения** – это свойство декомпозиции, которая вызывает генерацию ложных строк при обратном соединении декомпозированных отношений. **Чтобы не возникало зависимостей соединения, необходимо отношение приводить к 5НФ.**

**Отношение в 5НФ** – это отношение **без зависимостей соединения.**

N_объекта	Мебель	N_поставщика
31	Стол	P1
31	Стул	P2
52	Стул	P3
52	Кровать	P1

**Для того, чтобы отношение удовлетворяло 5-ой НФ, необходимо его разбить на следующие отношения:**

Объект – Мебель (N\_объекта, Мебель)

Поставщик – Мебель (N\_поставщика, Мебель)

Объекта - Поставщик (N\_объекта, N\_поставщика)

## СТАДИЯ АНОМАЛИИ ИЗБЫТОЧНОСТИ

НЕНОРМАЛИЗОВАННАЯ (НУЛЕВАЯ) ФОРМА (UNF)	Это состояние перед любой нормализацией. В таблице <b>присутствуют избыточные и сложные значения</b>
ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА (1NF)	Разбиваются <b>повторяющиеся и сложные значения</b> ; все экземпляры становятся <b>атомарными</b>
ВТОРАЯ НОРМАЛЬНАЯ ФОРМА (2NF)	Частичные зависимости разделяются на новые таблицы. Все строки <b>функционально зависимы от первичного ключа</b>
ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА (3NF)	Транзитивные зависимости разбиваются на новые <b>таблицы</b> . Не ключевые атрибуты зависят от первичного ключа
НОРМАЛЬНАЯ ФОРМА БОЙСА-КОДА (BCNF)	Транзитивные и частичные <b>функциональные зависимости</b> для всех <b>потенциальных ключей</b> разбиваются на новые таблицы
ЧЕТВЕРТАЯ НОРМАЛЬНАЯ ФОРМА (4NF)	Удаляются <b>многозначные зависимости</b>
ПЯТАЯ НОРМАЛЬНАЯ ФОРМА (5NF)	Удаляются <b>JOIN-зависимости</b> (зависимости соединения)

# ЭТАПЫ НОРМАЛИЗАЦИИ БАЗЫ ДАННЫХ

**Шаг 1 (приведение к 1НФ).** Задаются одно или несколько отношений, отображающих понятия предметной области. По модели предметной области (не по внешнему виду полученных отношений!) выписываются обнаруженные функциональные зависимости. Все отношения автоматически находятся в 1НФ.

**Шаг 2 (приведение к 2НФ).** Если в некоторых отношениях обнаружена зависимость атрибутов от части сложного ключа, то проводим декомпозицию этих отношений на несколько отношений согласно процедуре приведения к 2НФ.

**Шаг 3 (приведение к 3НФ).** Если в некоторых отношениях обнаружена зависимость некоторых неключевых атрибутов других неключевых атрибутов, то проводим декомпозицию этих отношений согласно процедуре приведения к 3НФ.

# ДЕНОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

**Нормализация — это не процесс целенаправленного приведения базы данных к какой-то определенной нормальной форме, нормализация — это набор принципов, зная и следуя которым, можно спроектировать базу данных, структура которой будет гарантировать отсутствие определенного рода аномалий.**

Таким образом, нет требования, которое обязывало бы приводить базу данных к максимально возможной нормальной форме, например, к 5 или 6, и не нужно нормализовать базу данных только для того, чтобы она была нормализована. **Максимально нормализованная база данных — это плохая база данных.**

**Денормализация — намеренное приведение структуры базы данных в состояние, не удовлетворяющее требованиям нормализации.**

Денормализация обычно проводится путем добавления избыточных данных в таблицу, т.е. тех данных, которые по требованиям той или иной нормальной формы должны выноситься в отдельную таблицу.



## Преимущества денормализации:

**Ускорение чтения данных.** Дублирование и объединение данных позволяет избежать дорогостоящих операций объединения таблиц и значительно ускорить выполнение запросов.

**Упрощение запросов.** За счет денормализации данные становятся доступными в одной таблице, что упрощает написание запросов и снижает сложность разработки.

**Улучшение производительности системы.** Более быстрое выполнение запросов позволяет обрабатывать большее количество запросов в течение определенного времени, что повышает производительность системы.

## Недостатки денормализации:

**Увеличение объема данных.** Дублирование данных приводит к увеличению объема хранимых данных, что требует больше места на диске.

**Потеря целостности данных.** Дублирование данных может привести к потере целостности, если необходимые обновления не выполняются правильно.

**Повышенное время выполнения записи.** Обновление дублированных записей требует дополнительного времени и ресурсов, поэтому время выполнения операций записи может увеличиться.