В теории баз данных существует несколько нормальных форм, которые используются для организации данных и устранения избыточности. Вот основные из них:

1. **Первая нормальная форма (1НФ)**:
   * Каждая таблица должна иметь уникальный первичный ключ.
   * Все столбцы должны содержать только атомарные (неделимые) значения, и каждое значение должно быть уникальным.
   * Порядок строк в таблице не имеет значения.
2. **Вторая нормальная форма (2НФ)**:
   * Таблица должна быть в 1НФ.
   * Все неключевые атрибуты должны полностью зависеть от первичного ключа. Это означает, что не должно быть частичных зависимостей от первичного ключа.
3. **Третья нормальная форма (3НФ)**:
   * Таблица должна быть в 2НФ.
   * Все атрибуты должны зависеть только от первичного ключа, то есть не должно быть транзитивных зависимостей.
4. **Форма Бойса-Кодда (БКНФ)**:
   * Это усиленная версия 3НФ.
   * Для каждой нетривиальной функциональной зависимости X → Y, X должен быть суперключом.
5. **Четвертая нормальная форма (4НФ)**:
   * Таблица должна быть в БКНФ.
   * Не должно быть многозначных зависимостей, то есть один атрибут не должен определять несколько независимых фактов.
6. **Пятая нормальная форма (5НФ)**:
   * Таблица должна быть в 4НФ.
   * Не должно быть зависимостей соединений, то есть данные не должны быть разделены на несколько таблиц таким образом, чтобы их объединение давало новую информацию.

Эти нормальные формы помогают проектировать базы данных, минимизируя избыточность и обеспечивая целостность данных.

**1НФ**

Атомарные (неделимые) значения в контексте баз данных означают, что каждое значение в столбце таблицы не может быть разделено на более мелкие части. Это значит, что каждое поле должно содержать только одно значение, а не набор значений.

### Пример:

Рассмотрим таблицу, которая хранит информацию о студентах и их хобби:

**Не в 1НФ (значения не атомарные):**

| **StudentID** | **Name** | **Hobbies** |
| --- | --- | --- |
| 1 | Иван | Чтение, Плавание |
| 2 | Мария | Рисование, Йога |

В этом примере столбец "Hobbies" содержит несколько значений (например, "Чтение, Плавание"), что нарушает требование атомарности.

**В 1НФ (значения атомарные):**

Чтобы привести таблицу к 1НФ, нужно разделить многозначные данные на отдельные строки:

| **StudentID** | **Name** | **Hobby** |
| --- | --- | --- |
| 1 | Иван | Чтение |
| 1 | Иван | Плавание |
| 2 | Мария | Рисование |
| 2 | Мария | Йога |

Теперь каждое значение в столбце "Hobby" является атомарным, и таблица находится в первой нормальной форме.

**2НФ**

Вторая нормальная форма (2НФ) требует, чтобы таблица была в первой нормальной форме (1НФ) и чтобы все неключевые атрибуты полностью зависели от первичного ключа. Это означает, что не должно быть частичных зависимостей от первичного ключа.

### Пример:

Рассмотрим таблицу, которая хранит информацию о заказах и продуктах:

**Не в 2НФ (частичные зависимости):**

| **OrderID** | **ProductID** | **ProductName** | **Price** | **Quantity** |
| --- | --- | --- | --- | --- |
| 1 | 101 | Молоко | 50 | 2 |
| 1 | 102 | Хлеб | 30 | 1 |
| 2 | 101 | Молоко | 50 | 3 |

В этом примере первичный ключ — это композитный ключ (OrderID, ProductID). Однако, ProductName и Price зависят только от ProductID, а не от всего первичного ключа (OrderID, ProductID). Это нарушает требования 2НФ.

**В 2НФ (устранение частичных зависимостей):**

Чтобы привести таблицу ко второй нормальной форме, нужно разделить данные на две таблицы:

**Таблица "Orders" (Заказы):**

| **OrderID** | **ProductID** | **Quantity** |
| --- | --- | --- |
| 1 | 101 | 2 |
| 1 | 102 | 1 |
| 2 | 101 | 3 |

**Таблица "Products" (Продукты):**

| **ProductID** | **ProductName** | **Price** |
| --- | --- | --- |
| 101 | Молоко | 50 |
| 102 | Хлеб | 30 |

Теперь каждая таблица находится во второй нормальной форме, так как все неключевые атрибуты полностью зависят от первичного ключа.

**3НФ**

Третья нормальная форма (3НФ) требует, чтобы таблица была во второй нормальной форме (2НФ) и чтобы все атрибуты зависели только от первичного ключа, то есть не должно быть транзитивных зависимостей. Это означает, что неключевые атрибуты не должны зависеть от других неключевых атрибутов.

### Пример:

Рассмотрим таблицу, которая хранит информацию о студентах и их курсах:

**Не в 3НФ (транзитивные зависимости):**

| **StudentID** | **StudentName** | **CourseID** | **CourseName** | **InstructorName** |
| --- | --- | --- | --- | --- |
| 1 | Иван | 101 | Математика | Петров |
| 2 | Мария | 102 | Физика | Сидоров |
| 3 | Анна | 101 | Математика | Петров |

В этом примере первичный ключ — это StudentID. Однако, CourseName и InstructorName зависят от CourseID, а не от StudentID. Это создает транзитивную зависимость.

**В 3НФ (устранение транзитивных зависимостей):**

Чтобы привести таблицу к третьей нормальной форме, нужно разделить данные на три таблицы:

**Таблица "Students" (Студенты):**

| **StudentID** | **StudentName** |
| --- | --- |
| 1 | Иван |
| 2 | Мария |
| 3 | Анна |

**Таблица "Courses" (Курсы):**

| **CourseID** | **CourseName** | **InstructorName** |
| --- | --- | --- |
| 101 | Математика | Петров |
| 102 | Физика | Сидоров |

**Таблица "Enrollments" (Зачисления):**

| **StudentID** | **CourseID** |
| --- | --- |
| 1 | 101 |
| 2 | 102 |
| 3 | 101 |

Теперь каждая таблица находится в третьей нормальной форме, так как все неключевые атрибуты зависят только от первичного ключа, и транзитивные зависимости устранены.

**Форма Бойса-Кодда (БКНФ)**

Форма Бойса-Кодда (БКНФ) — это усиленная версия третьей нормальной формы (3НФ). В БКНФ для каждой нетривиальной функциональной зависимости *X*→*Y*, *X* должен быть суперключом. Это означает, что любой детерминант должен быть кандидатом на роль ключа.

### Пример:

Рассмотрим таблицу, которая хранит информацию о проектах и сотрудниках, работающих над ними:

**Не в БКНФ (нарушение условия суперключа):**

| **ProjectID** | **EmployeeID** | **EmployeeName** | **Department** |
| --- | --- | --- | --- |
| 1 | 101 | Иван | IT |
| 1 | 102 | Мария | HR |
| 2 | 101 | Иван | IT |

В этом примере первичный ключ — это композитный ключ (ProjectID, EmployeeID). Однако, EmployeeName и Department зависят только от EmployeeID, а не от всего первичного ключа. Это нарушает требования БКНФ, так как EmployeeID не является суперключом для всей таблицы.

**В БКНФ (устранение нарушений):**

Чтобы привести таблицу к форме Бойса-Кодда, нужно разделить данные на две таблицы:

**Таблица "ProjectAssignments" (Назначения на проекты):**

| **ProjectID** | **EmployeeID** |
| --- | --- |
| 1 | 101 |
| 1 | 102 |
| 2 | 101 |

**Таблица "Employees" (Сотрудники):**

| **EmployeeID** | **EmployeeName** | **Department** |
| --- | --- | --- |
| 101 | Иван | IT |
| 102 | Мария | HR |

Теперь каждая таблица находится в форме Бойса-Кодда, так как все детерминанты (EmployeeID в таблице "Employees") являются суперключами.

**4НФ**

Четвертая нормальная форма (4НФ) требует, чтобы таблица была в форме Бойса-Кодда (БКНФ) и чтобы не было многозначных зависимостей. Многозначная зависимость возникает, когда два независимых атрибута определяют многозначное соответствие.

### Пример:

Рассмотрим таблицу, которая хранит информацию о студентах, их курсах и клубах, в которых они состоят:

**Не в 4НФ (многозначные зависимости):**

| **StudentID** | **Course** | **Club** |
| --- | --- | --- |
| 1 | Math | Chess |
| 1 | Physics | Chess |
| 1 | Math | Football |
| 2 | Math | Swimming |

В этом примере StudentID определяет многозначные зависимости между Course и Club. Это означает, что для одного StudentID может быть несколько курсов и несколько клубов, и они не зависят друг от друга.

**В 4НФ (устранение многозначных зависимостей):**

Чтобы привести таблицу к четвертой нормальной форме, нужно разделить данные на две таблицы:

**Таблица "StudentCourses" (Курсы студентов):**

| **StudentID** | **Course** |
| --- | --- |
| 1 | Math |
| 1 | Physics |
| 2 | Math |

**Таблица "StudentClubs" (Клубы студентов):**

| **StudentID** | **Club** |
| --- | --- |
| 1 | Chess |
| 1 | Football |
| 2 | Swimming |

Теперь каждая таблица находится в четвертой нормальной форме, так как многозначные зависимости устранены.

**5НФ**

Пятая нормальная форма (5НФ), также известная как проектно-соединительная нормальная форма (PJ/NF), требует, чтобы таблица была в четвертой нормальной форме (4НФ) и чтобы не было зависимостей соединений. Зависимость соединения возникает, когда информация в таблице может быть восстановлена путем соединения нескольких таблиц.

### Пример:

Рассмотрим таблицу, которая хранит информацию о поставщиках, проектах и частях, которые они поставляют для этих проектов:

**Не в 5НФ (зависимости соединений):**

| **Supplier** | **Project** | **Part** |
| --- | --- | --- |
| A | P1 | X |
| A | P2 | Y |
| B | P1 | X |
| B | P2 | Z |

В этом примере таблица содержит зависимости соединений. Например, можно определить, что поставщик A поставляет часть X для проекта P1, но также можно узнать, что поставщик B поставляет часть X для проекта P1, что создает зависимость соединения.

**В 5НФ (устранение зависимостей соединений):**

Чтобы привести таблицу к пятой нормальной форме, нужно разделить данные на три таблицы:

**Таблица "SupplierProjects" (Поставщики и проекты):**

| **Supplier** | **Project** |
| --- | --- |
| A | P1 |
| A | P2 |
| B | P1 |
| B | P2 |

**Таблица "ProjectParts" (Проекты и части):**

| **Project** | **Part** |
| --- | --- |
| P1 | X |
| P2 | Y |
| P2 | Z |

**Таблица "SupplierParts" (Поставщики и части):**

| **Supplier** | **Part** |
| --- | --- |
| A | X |
| A | Y |
| B | X |
| B | Z |

Теперь каждая таблица находится в пятой нормальной форме, так как зависимости соединений устранены. Информация о том, какие части поставляются для каких проектов, может быть восстановлена путем соединения этих таблиц.