Дайте определение 1НФ, 2НФ, 3НФ, НФБК, 4НФ

**Первая нормальная форма (1НФ)**

Таблица находится в первой нормальной форме, если:

1. Все атрибуты содержат атомарные (неделимые) значения.
2. В таблице нет повторяющихся строк.
3. Каждый столбец содержит значения только одного типа данных.

**Вторая нормальная форма (2НФ)**

Таблица находится во второй нормальной форме, если:

1. Она уже находится в 1НФ.
2. Все неключевые атрибуты полностью зависят от всего составного ключа (если ключ составной). Это означает, что не должно быть частичных зависимостей, когда неключевой атрибут зависит только от части составного ключа.

**Третья нормальная форма (3НФ)**

Таблица находится в третьей нормальной форме, если:

1. Она уже находится во 2НФ.
2. Никакой неключевой атрибут не зависит транзитивно от ключа. Это означает, что каждый неключевой атрибут должен зависеть только от первичного ключа и не должен зависеть от других неключевых атрибутов.

**Нормальная форма Бойса-Кодда (НФБК)**

Таблица находится в нормальной форме Бойса-Кодда, если:

1. Она уже находится в 3НФ.
2. Для любой функциональной зависимости X -> Y, X является суперклюзом, то есть множество X является или содержит ключ таблицы. НФБК более строгая форма 3НФ и решает некоторые аномалии, которые 3НФ может не устранить.

**Четвертая нормальная форма (4НФ)**

Таблица находится в четвертой нормальной форме, если:

1. Она уже находится в НФБК.
2. В таблице нет многозначных зависимостей. Многозначная зависимость возникает, когда один атрибут в таблице определяет множество значений другого атрибута независимо от других атрибутов.

Назовите основные правила перехода от концептуальной к логической реляционной схеме.

Переход от концептуальной модели данных к логической реляционной схеме включает преобразование высокоуровневых концептуальных элементов в реляционные структуры, которые могут быть реализованы в СУБД. Основные шаги этого процесса включают:

**1. Идентификация сущностей и атрибутов**

* **Сущности** концептуальной модели становятся таблицами в реляционной схеме.
* **Атрибуты** сущностей становятся столбцами этих таблиц.
* Определяются **первичные ключи** (Primary Keys) для каждой таблицы.

**2. Обработка связей (отношений)**

* **Связи один-к-одному (1:1)** могут быть реализованы как объединение двух сущностей в одну таблицу или путем добавления внешнего ключа (Foreign Key) из одной таблицы в другую.
* **Связи один-ко-многим (1:M)** реализуются путем добавления внешнего ключа в таблицу, представляющую "много".
* **Связи многие-ко-многим (M:N)** реализуются созданием промежуточной таблицы (связующей таблицы), которая содержит внешние ключи, ссылающиеся на первичные ключи обоих связанных таблиц.

**3. Нормализация данных**

* Применяются правила нормализации (1НФ, 2НФ, 3НФ и, при необходимости, НФБК и 4НФ) для устранения избыточности и предотвращения аномалий обновления.
* Таблицы разделяются или объединяются, чтобы обеспечить соблюдение нормальных форм.

**4. Определение доменов данных**

* Определяются **типы данных** для каждого столбца в соответствии с атрибутами концептуальной модели (например, INT, VARCHAR, DATE и т.д.).
* Задаются **ограничения целостности** (например, NOT NULL, UNIQUE).

**5. Уточнение ключей и ограничений**

* Уточняются **первичные ключи** для каждой таблицы.
* Определяются **внешние ключи** для установления и поддержания связей между таблицами.
* Добавляются другие **ограничения целостности** (например, CHECK, DEFAULT).

**6. Управление производительностью**

* Рассматривается **денормализация** для повышения производительности, если это необходимо, при сохранении целостности данных.
* Планируются **индексы** для ускорения поиска и соединений.

Как в реляционной модели реализуются супертипы и подтипы?

В реляционной модели супертипы и подтипы могут быть реализованы с помощью методов, таких как:

1. **Единая таблица** (Single Table Inheritance)
2. **Отдельные таблицы для каждого подтипа** (Class Table Inheritance)
3. **Смешанный подход** (Hybrid Approach)

**Единая таблица (Single Table Inheritance)**

* Все атрибуты супертипа и подтипов хранятся в одной таблице.
* Используется дополнительный столбец для идентификации типа записи.
* Столбцы, не применимые к определенному подтипу, могут содержать значения NULL.

**Отдельные таблицы для каждого подтипа (Class Table Inheritance)**

* Создаются отдельные таблицы для каждого подтипа.
* Каждая таблица содержит атрибуты подтипа и ссылку на супертип.
* Позволяет избежать NULL-значений и более гибко управлять атрибутами подтипов.

**Смешанный подход (Hybrid Approach)**

* Комбинирует преимущества двух предыдущих методов.
* Некоторые атрибуты хранятся в общей таблице супертипа, а специфичные атрибуты хранятся в отдельных таблицах для каждого подтипа.