1. **Какие требования у первой нормальной формы?**

Первая нормальная форма сводится к трем правилам:

1. Каждая ячейка таблицы может хранить только одно значение
2. Все данные в одной колонке могут быть только одного типа
3. Каждая запись в таблице должна однозначно отличаться от других записей
4. **Какие требования у второй нормальной формы?**

Вторая нормальная форма (2NF) — одна из возможных нормальных форм отношения в реляционной базе данных.

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от (каждого) её потенциального ключа.

1. **Какие требования у третьей нормальной формы?**

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

1. **Какие требования у четвёртой нормальной формы?**

Переменная отношения R находится в четвёртой нормальной форме, если она находится в НФБК и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от её потенциальных ключей.

*Нормальная форма Бойса-Кодда (НФБК):* отношение находится в НФБК тогда и только тогда, когда каждый его детерминант является потенциальным ключом.

1. **Какие преимущества и недостатки у нормализации?**

*Преимущества нормализации:*

* 1. *Устранение избыточности даны* **(**Одним из основных преимуществ нормализации является устранение избыточности данных. Когда данные дублируются в разных частях базы данных, это может привести к проблемам согласованности и целостности данных. Нормализация позволяет избежать таких проблем, разделяя данные на отдельные таблицы и связывая их с помощью отношений.)
  2. *Улучшение производительности* **(**Нормализация может улучшить производительность базы данных. Поскольку данные хранятся в более компактной и структурированной форме, запросы к базе данных могут выполняться быстрее. Кроме того, нормализация позволяет эффективно использовать индексы и другие методы оптимизации запросов.)
  3. *Упрощение обновления данных* **(**Когда данные хранятся в нормализованной форме, обновление данных становится проще и безопаснее. Изменения нужно вносить только в одной таблице, а не в нескольких местах. Это уменьшает вероятность ошибок и обеспечивает согласованность данных.)
  4. *Улучшение структуры базы данных* **(**Нормализация помогает создать логическую и структурированную модель базы данных. Это делает базу данных более понятной и легкой для администрирования. Кроме того, нормализация позволяет легко добавлять, изменять и удалять данные без необходимости вносить сложные изменения в структуру базы данных.)
  5. *Повышение надежности и целостности данных* **(**Нормализация способствует повышению надежности и целостности данных. Поскольку данные хранятся в отдельных таблицах, существует меньше вероятность ошибок при обновлении и удалении данных. Кроме того, нормализация позволяет определить ограничения целостности, которые обеспечивают правильность и согласованность данных.)

*Недостатки нормализации*

* + 1. *Избыточность запросов (*При нормализации данных может потребоваться выполнение большого количества запросов для получения полной информации. Это может привести к увеличению времени выполнения запросов и снижению производительности системы.)
    2. *Сложность проектирования (*Нормализация требует более тщательного и сложного проектирования базы данных. Необходимо определить правильные связи между таблицами и правильно разделить данные. Это может быть сложно для новичков в области баз данных.)
    3. *Увеличение объема хранилища (*Нормализация может привести к увеличению объема хранилища данных. Поскольку данные разделены на несколько таблиц, может потребоваться больше места для их хранения. Это может быть проблемой, особенно при работе с большими объемами данных.)
    4. *Сложность обновления данных (*При нормализации данных обновление информации может быть сложным. Если данные хранятся в нескольких таблицах, необходимо обновлять информацию в каждой из них. Это может быть трудоемким и вызывать проблемы с целостностью данных.)
    5. *Потеря производительности при объединении таблиц (*При выполнении запросов, которые требуют объединения нескольких таблиц, может возникнуть потеря производительности. Объединение таблиц может быть затратным с точки зрения времени выполнения запроса, особенно если таблицы содержат большое количество данных.)

1. **Какие преимущества у модели "сущность-связь" перед реляционной моделью?**

Преимущества модели "сущность-связь" перед реляционной моделью включают:

1. Интуитивность (Модель "сущность-связь" легко понять и использовать из-за своей интуитивной природы. Она имеет графическую представления различных сущностей и связей между ними, что облегчает понимание и визуализацию данных.)
2. Более высокий уровень абстракции (Модель "сущность-связь" позволяет абстрагироваться от конкретных таблиц и структуры базы данных, и позволяет разработчикам и аналитикам сфокусироваться на бизнес-логике и логическом моделировании данных, вместо работы с таблицами и отношениями.)
3. Правильное исчисление множественности (Модель "сущность-связь" позволяет точно определить отношения и связи между сущностями, включая определение кардинальности и множественности связей. Это помогает избежать ошибок или неоднозначности в структуре данных.)
4. Лучшая гибкость (Модель "сущность-связь" обладает большей гибкостью при изменении структуры данных. Она позволяет добавлять новые сущности или связи без изменения базы данных, что облегчает поддержку и развитие системы.)
5. Визуализация сложных отношений (Модель "сущность-связь" облегчает визуализацию и анализ сложных отношений и иерархий в данных. Это может быть полезно для проектирования и оптимизации базы данных или для понимания сложных бизнес-процессов.)
6. Хороший стартовый пункт для реляционных баз данных (Модель "сущность-связь" часто используется в качестве стартовой точки для разработки реляционной базы данных. Она помогает логически организовать данные и определить ключевые сущности и связи перед тем как переходить к реляционной моделированию.)
7. **Какие основные элементы использует модель "сущность-связь" для описания предметной области?**

Основными элементами модели "сущность-связь" являются:

1. Сущность (Entity): Представляет отдельный объект или понятие в предметной области. Сущности имеют атрибуты, которые описывают их свойства или характеристики.
2. Связь (Relationship): Представляет отношение или связь между двумя или более сущностями. Связи описывают, как одна сущность связана с другой и могут иметь свои атрибуты.
3. Атрибут (Attribute): Представляет собой характеристику или свойство сущности или связи. Атрибуты описывают данные, которые связаны со сущностью или связью и могут иметь типы данных (например, число, строка, дата и т.д.).
4. Ключ (Key): Представляет уникальный идентификатор сущности. Ключ может состоять из одного или нескольких атрибутов и используется для однозначного идентификации каждого экземпляра сущности.
5. Кардинальность (Cardinality): Определяет количество экземпляров сущности, которые могут быть связаны через связь. Например, один-к-одному, один-ко-многим, многие-ко-многим.
6. Ассоциация (Association): Представляет связь между двумя или более сущностями, которая не имеет своих атрибутов или ролей. Ассоциации могут быть использованы для описания взаимодействий или взаимосвязей между сущностями.
7. **Что обозначает наследование в расширенной модели "сущность-связь"?**

В расширенной модели “сущность-связь” наследование обозначает процесс, при котором атрибуты и свойства одной сущности (суперкласса или родительской сущности) могут быть унаследованы другой сущностью (подклассом или дочерней сущностью). Это позволяет создавать иерархию связанных между собой объектов, где каждый объект может иметь общие и специфические свойства, в зависимости от его положения в иерархии. Наследование также позволяет сократить избыточность данных и упростить манипуляции с данными.

1. **Как преобразуется модель "сущность-связь" в реляционную модель?**

При преобразовании в реляционную модель каждая сущность превращается в таблицу, где каждый атрибут становится столбцом, а каждая запись представляет экземпляр сущности. Ключевым атрибутом сущности обычно становится первичным ключом таблицы.

Связи между сущностями становятся отношениями или таблицами, которые образуются путем соединения таблиц с помощью внешних ключей. Если связь между сущностями имеет атрибуты, то эти атрибуты также становятся столбцами в таблице отношения.

Таким образом, преобразование модели "сущность-связь" в реляционную модель сводится к созданию таблиц для каждой сущности и связи, а также установлению отношений между этими таблицами с помощью внешних ключей.

1. **Можно ли использовать модель "сущность-связь" не для проектирования баз данных?**

Она также может быть полезна при разработке объектно-ориентированных систем, анализе и моделировании бизнес-процессов, проектировании интерфейсов и других задачах, где требуется структурирование и организация информации.