1. **Реляционные базы данных** отличаются от **нереляционных** следующим образом:
   * **Структура данных**: В реляционных базах данных данные организованы в виде таблиц (отношений), где каждая строка представляет собой запись, а каждый столбец - атрибут. В нереляционных базах данных данные могут быть организованы иначе, например, в виде документов, графов или ключ-значение.
   * **Схема данных**: Реляционные базы данных имеют строго определенную схему данных, которая описывает структуру таблиц и связи между ними. В нереляционных базах данных схема может быть более гибкой и динамической.
   * **Язык запросов**: Реляционные базы данных используют язык SQL для выполнения запросов. В нереляционных базах данных используются различные языки запросов в зависимости от типа базы данных (например, MongoDB использует язык запросов BSON).
   * **Транзакции и целостность данных**: Реляционные базы данных обеспечивают поддержку транзакций и обеспечивают целостность данных с помощью ограничений (например, уникальность значений). В нереляционных базах данных это может быть менее строго регулируемо.
2. **Реляционная модель данных** - это способ организации данных в виде таблиц (отношений), где каждая строка представляет собой запись, а каждый столбец - атрибут. Отношения между таблицами устанавливаются с помощью ключей.
3. **Классификация СУБД**:
   * **Реляционные СУБД**: Основаны на реляционной модели данных. Примеры: MySQL, PostgreSQL, Oracle.
   * **Нереляционные СУБД**: Используют другие модели данных (например, документы, графы, ключ-значение). Примеры: MongoDB, Cassandra, Redis.
   * **Объектно-ориентированные СУБД**: Работают с объектами и классами. Пример: db4o.
   * **Интернет-ориентированные СУБД**: Оптимизированы для хранения и обработки данных веб-приложений. Пример: Firebase.
4. **Клиент-серверная СУБД** - это архитектура, в которой база данных разделена на две части: серверную (которая управляет данными) и клиентскую (которая обращается к серверу для доступа к данным).
5. **Ограничения целостности** - это правила, которые обеспечивают целостность данных в базе данных. Примеры ограничений: уникальность значений, ссылочная целостность (связи между таблицами), ограничения на значения атрибутов.
6. **Первичный ключ** - это уникальный идентификатор для каждой записи в таблице. Он обеспечивает уникальность записей и используется для связей между таблицами. Ограничение первичного ключа гарантирует, что значения ключа не повторяются.
7. **Внешний ключ** - это атрибут в таблице, который ссылается на первичный ключ другой таблицы. Он используется для установления связей между таблицами. Ограничение внешнего ключа гарантирует, что значения внешнего ключа существуют в связанной таблице.