Так, вот тут будут расписаны прошлогодние рубежки, постараюсь расписать все понятно, кратко и без ошибок

1 вариация прошлогодней рубежки

Ссылка: https://www.notion.so/555833bc79984a33980d91b371a6b0ae

Вариант 1

1. ***Типовые функции информационной системы:***
   - Функции сбора и регистрации информационных ресурсов
   - Функции хранения информационных ресурсов
   - Функции актуализации информационных ресурсов
   - Функции обработки информационных ресурсов
   - Функции предоставления информационных ресурсов пользователям
2. ***Отличия иерархической модели данных от сетевой:***
   - *Иерархическая модель:* связи "один-к-многим", один родитель на узел и несколько детей
   - *Сетевая модель:* связи "многие-ко-многим", представляется в виде графа, может быть несколько родителей и детей для одного узла, обеспечивается целостность, так как нет дублирования данных из-за поддержки более гибких связей
3. ***Типы атрибутов:***
   - *Простые атрибуты* - не делятся на подчасти
     Пример: ФИО - не делится, так как одна строка цельная
   - *Составные атрибуты* - состоят из нескольких атрибутов
     Пример: Адрес - можно разделить на Улицу, Номер дома, Город и т.д.
   - *Идентифицирующие атрибуты (Обязательные атрибуты)* - однозначно идентифицируют сущность
     Пример: Номер ИСУ - уникален для каждого студента
   - *Необязательные атрибуты* — могут не иметь значения для некоторых записей и могут быть NULL
     Пример: Отчество - может просто не быть у некоторых студентов
   - *Однозначные атрибуты* - хранят только одно значение
     Пример: Дата рождения - у каждого студента только одна дата рождения
   - *Многозначные атрибуты* - могут содержать несколько значений
     Пример: Телефонные номера - у студента может быть несколько телефонных номеров
4. ***Свойство отношений:***

- *Уникальность имени таблицы* - каждая таблица имеет уникальное имя в БД, что позволяет однозначно ее идентифицировать
- *Уникальность кортежей и атрибутов*- каждый кортеж и атрибут в таблице уникален, что исключает возможность дублирования данных
- *Неупорядоченность кортежей и атрибутов* - порядок атрибутов и кортежей не имеет значения, так как данные идентифицируются по именам атрибутов и извлекаются независимо от порядка
- *Атомарность значений* - каждая ячейка в таблице содержит одно неделимое значение
- *Однородность доменов* - значения атрибутов берутся из одного и того же домена (типа данных), чтобы сохранить единообразие
- *Целостность данных* - значения атрибутов должны соответствовать определенным правилам и ограничениям, например, `NOT NULL` для обязательных полей или ограничение уникальности для первичного ключа

5. ***Оператор `SELECT` и порядок выполнения операторов:***
   `SELECT` - оператор, который используется для извлечения данных из одной или нескольких таблиц, он позволяет указать, какие поля вернуть, какие условия применить и в каком порядке вывести данные

   Порядок выполнения операторов:
   1. `FROM`
   2. `JOIN`
   3. `WHERE`
   4. `GROUP BY`
   5. `HAVING`
   6. `SELECT`
   7. `DISTINCT`
   8. `ORDER BY`
   9. `LIMIT`
6. ***Определение БД:***
   База данных - набор постоянно хранимых данных, используемых прикладными системами какого-либо предприятия
7. ***Уровни архитектуры ANSII/Sparc:***
   ANSII/Sparc - это стандарты подходов к построению архитектуры БД:
   1. *Внутренний уровень* - отвечает за физическое хранение данных на диске и оптимизацию работы с ними. Скрыт от пользователя, влияет на производительность и безопасность
   2. *Концептуальный уровень* - логическая структура данных (сущности, атрибуты, связи), с которой работают разработчики при проектировании базы

3. *Внешний уровень* - представление данных для пользователей, показывающее нужную информацию и скрывающее технические детали хранения

8. ***Определение тета-соединения:***
   Тета-соединение — определяет отношение, содержащее кортежи из декартового произведения `R x S`, удовлетворяющие предикату `F = R.a_i θ S.b_j`, где `θ` — одна из операций сравнения `(>, <, =, ...)`.
9. ***Определение операции "проекция":***
   Проекция `П_{a1, ..., an}(R)` — результатом является новое отношение, содержащее вертикальное подмножество исходного отношения, создаваемое посредством извлечения указанных атрибутов и исключения из результата дублирующихся атрибутов.
10. ***В чём отличие дублирования от избыточности?***
    Отличие заключается в том, что *избыточность* подразумевает, что есть возможность восстановить или получить данные на основе других записей, но при это они все равно хранятся повторно, хотя их можно было бы извлечь из других данных. В случае дублирования такой возможности нет - это просто повторение данных, которые никак не зависят от других записей не могут быть восстановлены из них.

    Дублирование — это прямое повторение одних и тех же данных в разных местах (например, хранение имени студента в нескольких таблицах), что может привести к несоответствиям при обновлении

    Избыточность — более широкое понятие, которое включает дублирование, но также охватывает излишние данные, которые не обязательны для уникальной идентификации объекта

Пример избыточности:
Таблица "Студенты"

ФИО	**Номер группы**	**Факультет**
Иванов И.И.	М3210	ФИТИП
Петров П.П.	М3211	ФИТИП
Сидоров С.С.	М3212	ФИТИП

Таблица "Группы"

Номер группы	**Факультет**
М3210	ФИТИП
М3211	ФИТИП

Номер группы	Факультет
М3212	ФИТИП

В этом случае информация о факультете хранится как отдельный атрибут у каждого студента, хотя её можно получить, исходя из номера группы

Пример дублирования:
Таблица "Студенты" с дублированием

ФИО	Номер группы	Факультет	Форма обучения
Иванов И.И.	М3210	ФИТИП	Контракт
Петров П.П.	М3211	ФИТИП	Бюджет
Сидоров С.С.	М3212	ФИТИП	Контракт

В этом случае информация о факультете повторяется для каждого студента, принадлежащего одной и той же группе, но её нельзя восстановить из других данных — это просто дублирование значений, так как факультет зависит от группы, а не от каждого студента

11. ***Функциональная зависимость:***
    Функциональная зависимость между атрибутами - это связь, при которой каждому значению атрибута *X* соответствует ровно одно значение атрибута *Y*.

$$R : X \to Y$$

12. ***Когда отношение в 1-ой Нормальной Форме:***
    1-я Нормальная Форма - таблица находится в **первой нормальной форме**, если все ее атрибуты являются атомарными, то есть неделимыми
13. ***Степень отношения:***
    Степень отношения - это количество его атрибутов. Степень отношения показывает, сколько различных характеристик описывает объекты, которые есть в таблице

Вариант 2:

1. ***Что описывается на верхнем уровне архитектуры ANSI/SPARC?***
   Внешний уровень - представление данных для пользователей, показывающее нужную информацию и скрывающее технические детали хранения. Этот уровень абстракции про то, как принимать эффективные решения на основе данных
2. ***Определение СУБД:***
   СУБД - программное обеспечение с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а так же осуществлять к ней контролируемый доступ

3. ***Типы связей в модели сущность связь:***
    - Один-к-одному
    - Один-ко-многим
    - Многие-ко-многим
4. ***В чем отличие реляционной и постреляционной моделей данных?***
    - *Реляционная модель данных* требует, чтобы каждое поле таблицы содержало неделимое (атомарное) значение. Связи между таблицами задаются через общие атрибуты, и структура данных строго фиксирована
    - *Постреляционная модель данных* позволяет хранить сложные данные в одном поле, например, массивы или JSON, что удобно для записей с разными атрибутами. Это снижает разряженность таблиц, но усложняет проверку данных и может замедлять поиск
5. ***Что такое кардинальность отношения в реляционной модели данных?***
   Кардинальность отношения - это количество кортежей, которое содержит отношение
6. ***Дайте определение операции "Выборка":***

   $$\textbf{Выборка } \sigma_{\text{предикат}}(R) -$$

   Операция, результатом которой является отношение, которое содержит только те кортежи из исходного отношения, которые удовлетворяют заданному условию (предикату)
7. ***Дайте определение операции "Естественное соединение":***

   $$\textbf{Естественное соединение } R \bowtie S$$

   Операция, которая выполняет соединение по эквивалентности двух отношений по всем общим атрибутам, из результатов которого исключили по одному экземпляру каждого атрибута
8. ***Дайте определение аномалии модификации:***
   Изменение данных в одной записи может потребовать внесения изменений в другие записи

```
Пример: Если изменить название факультета для группы M3212, то необходимо
внести изменения для всех студентов этой группы. Если не внести их для
одного из студентов, данные окажутся несогласованными
```

9. ***Порядок команд в SQL для написании запросов и порядок их выполнения:***
   Порядок команд при написании SQL-запросов:

    1. **SELECT**
    2. **DISTINCT**
    3. **FROM**
    4. **JOIN**

5. **WHERE**
6. **GROUP BY**
7. **HAVING**
8. **ORDER BY**
9. **LIMIT**

Порядок их выполнения:

1. **FROM**
2. **JOIN**
3. **WHERE**
4. **GROUP BY**
5. **HAVING**
6. **SELECT**
7. **DISTINCT**
8. **ORDER BY**
9. **LIMIT**

10. ***Какие условия должны быть, чтобы отношение находилось во второй нормальной форме?***
    2-я Нормальная Форма - таблица находится во **второй нормальной форме**, если находится в первой нормальной форме и все неключевые атрибуты функционально полностью зависят от первичного ключа
11. ***Суперключ:***
    Суперключ - это атрибут (множество атрибутов), который единственным, уникальным образом идентифицирует кортеж.
12. ***Частичная функциональная зависимость:***
    Частичная функциональная зависимость - зависимость от части составного ключа

Вариант 3:

1. ***Определение "Данные":***
   - *Данные* - это отдельные факты или значения без какого-либо контекста, которые сами по себе не имеют смысла (из инета)
   - *Данные* — это поддающиеся многократной интерпретации представления информации в формализованном виде, пригодном для передачи, интерпретации и обработки (Маятин)
2. ***Что такое полная функциональная зависимость?***
   Полная функциональная зависимость - явление, когда неключевой атрибут зависит от всего составного ключа
3. ***Концептуальный уровень ANSI/SPARC:***
   Концептуальный уровень - логическая структура данных (сущности, атрибуты, связи), с которой работают разработчики при проектировании базы

4. ***В чем отличие между концептуальной и логической модели данных?***
   Концептуальная модель данных отражает общее представление данных без привязки к техническим деталям, описывая основные сущности, их атрибуты и связи. Она фокусируется на том, какие данные важны и как они соотносятся

   Логическая модель данных конкретизирует концептуальную модель, добавляя структуру для реализации в СУБД. Она включает типы данных, ключи и ограничения, но остаётся независимой от конкретных технологий хранения данных
5. ***Определение "Домен":***
   Домен - это множество допустимых значений атрибутов
6. ***Что такое левое-внешнее соединение в реляционной алгебре?***
   Левое внешнее соединение $R \supset \triangleleft S$ - это естественное соединение, при котором в результирующее отношение включаются также кортежи отношения R, не имеющие совпадающих значений в общих атрибутах отношения S
7. ***Операция объединения:***
   Объединение двух отношений R и S определяет новое отношение, которое включает все кортежи, содержащиеся только в R, все кортежи, содержащиеся только в S и кортежи, содержащиеся и в R, и в S с исключением дубликатов
8. ***Третья нормальная форма:***
   3-я Нормальная Форма - таблица находится в **третьей нормальной форме**, если она находится во второй нормальной форме и все неключевые атрибуты независимы друг от друга и функционально полностью зависят от первичного ключа (отсутствуют транзитивные зависимости)
9. ***Алгоритм соединения таблиц вложенными циклами:***

```
type Row struct {
        Attr int
        Data string
}

func nestedLoopJoin(R, S []Row) []Row {
        var F []Row

        for _, r := range R {
                for _, s := range S {
                        if r.Attr == s.Attr {
                                joinedRow := Row{Attr: r.Attr, Data:
r.Data + " " + s.Data}
                                F = append(F, joinedRow)
                        }
                }
        }
```

```
        return F
}
```

10. ***Плюсы и минусы иерархической модели данных:***
    - **Плюсы**:
      - Удобна для восприятия человеком, так как соответствует реальным иерархиям
      - Простая структура, легко понимать и использовать для хранения иерархических данных
      - Быстро при транзакциях
    - **Минусы**:
      - Дублирование данных, что приводит к избыточности
      - Сложности в поддержании целостности данных и обновлении при изменении иерархии
      - Нельзя сделать связь "многие-ко-многим"
11. ***Определение "Потенциальный ключ":***
    Потенциальный ключ — это суперключ, который не содержит подмножества, также являющегося суперключом этого отношения
12. ***Определение "База данных":***
    База данных - набор постоянно хранимых данных, используемых прикладными системами какого-либо предприятия

Вариант 4:

1. ***В чем минусы хранения в файловой системе (файловом хранилище)?***
    - *Отсутствие структурированности* - данные хранятся в виде файлов без строгой структуры
    - *Ограниченные возможности поиска*
    - *Отсутствие транзакций и целостности данных* - невозможно гарантировать целостность данных и выполнять атомарные операции
    - *Отсутствие параллельной работы* - нет возможности одновременно работать нескольким пользователям с файлами из-за возможных ошибок
2. ***Определение "СУБД"***
    СУБД - программное обеспечение с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а так же осуществлять к ней контролируемый доступ
3. ***Внешний/внутренний ключ, супер ключ:***
    - *Внешний ключ* - это атрибут (множество атрибутов) внутри отношения, который соответствует потенциальному ключу некоторого (возможно того же самого) отношения
    - *Внутренний ключ* - атрибут, предназначенный для внутреннего использования в системе, например, технический идентификатор, который

не связан напрямую со значением данных **(!!!Не уверен, правильно ли это!!!)**
    - *Супер ключ* - это атрибут (множество атрибутов), который единственным образом идентифицирует кортеж
4. ***Что такое разность, полусоединение?***
    - *Разность R-S* - состоит из кортежей, которые есть в R, но отсутствуют в S. Разность двух отношений определена только если они совместны по объединению
    - *Полусоединение R $\triangleright$F S* - отношение, содержащее кортежи R, которые входят в тета-соединение R и S
5. ***Что такое транзитивная функциональная зависимость?***
   Транзитивная функциональная зависимость - зависимость через промежуточный атрибут (Y зависит от X, X не зависит от Y, Z зависит от Y), то зависимость A -> C - транзитивная
6. ***Внутренний уровень ANSII/SPARC:***
   Внутренний уровень - отвечает за физическое хранение данных на диске и оптимизацию работы с ними. Скрыт от пользователя, влияет на производительность и безопасность
7. ***Плюсы и минусы многомерной конструкции:***
    - *Плюсы:*
        - Быстрая агрегация и анализ данных
        - Поддержка сложных аналитических запросов
        - Удобна для сводок и отчетности
    - *Минусы:*
        - Высокий расход памяти
        - Трудности с обновлением данных в реальном времени
        - Ограничена в работе с детализированными данными
8. ***Определение "Кортеж"***
   Кортеж - отдельная строка в таблице
9. ***Виды связей в таблице:***
    - *Один-к-одному* - одна запись соответствует одной записи
    - *Один-ко-многим* - одна запись связана с несколькими записями
    - *Многие-ко-многим* - записи из одной таблицы могут быть связаны с несколькими записями в другой таблице через промежуточную таблицу
10. ***Аномалия добавления:***
    Невозможно добавить данные без указания дополнительных сведений, которые могут быть избыточными
11. ***Нормальная форма и нормализация:***
    Нормальные формы - это стандартизированные правила для организации данных в реляционных таблицах, которые помогают минимизировать избыточность и избежать аномалий

Нормализация - преобразования отношения к виду, отвечающему нормальной форме

12. ***Плюсы и минусы иерархической модели данных:***
    - **Плюсы**:
        - Удобна для восприятия человеком, так как соответствует реальным иерархиям
        - Простая структура, легко понимать и использовать для хранения иерархических данных
        - Быстро при транзакциях
    - **Минусы**:
        - Дублирование данных, что приводит к избыточности
        - Сложности в поддержании целостности данных и обновлении при изменении иерархии
        - Нельзя сделать связь "многие-ко-многим"

3 вариация прошлогодней рубежки

Ссылка: https://docs.google.com/document/d/12czJriBtfu75eq8bcXLX3mTt_hKFpLts/edit

Вариант 1:

1. ***Типовые функции информационной системы:***
    - Функции сбора и регистрации информационных ресурсов
    - Функции хранения информационных ресурсов
    - Функции актуализации информационных ресурсов
    - Функции обработки информационных ресурсов
    - Функции предоставления информационных ресурсов пользователям
2. ***Отличия иерархической модели данных от сетевой:***
    - *Иерархическая модель:* связи "один-к-многим", один родитель на узел и несколько детей
    - *Сетевая модель:* связи "многие-ко-многим", представляется в виде графа, может быть несколько родителей и детей для одного узла, обеспечивается целостность, так как нет дублирования данных из-за поддержки более гибких связей
3. ***Типы атрибутов:***
    - *Простые атрибуты* - не делятся на подчасти
      Пример: ФИО - не делится, так как одна строка цельная
    - *Составные атрибуты* - состоят из нескольких атрибутов
      Пример: Адрес - можно разделить на Улицу, Номер дома, Город и т.д.
    - *Идентифицирующие атрибуты (Обязательные атрибуты)* - однозначно идентифицируют сущность
      Пример: Номер ИСУ - уникален для каждого студента

- *Необязательные атрибуты* — могут не иметь значения для некоторых записей и могут быть NULL
  Пример: Отчество - может просто не быть у некоторых студентов
- *Однозначные атрибуты* - хранят только одно значение
  Пример: Дата рождения - у каждого студента только одна дата рождения
- *Многозначные атрибуты* - могут содержать несколько значений
  Пример: Телефонные номера - у студента может быть несколько телефонных номеров

4. ***Свойство отношений:***
   - *Уникальность имени таблицы* - каждая таблица имеет уникальное имя в БД, что позволяет однозначно ее идентифицировать
   - *Уникальность кортежей и атрибутов*- каждый кортеж и атрибут в таблице уникален, что исключает возможность дублирования данных
   - *Неупорядоченность кортежей и атрибутов* - порядок атрибутов и кортежей не имеет значения, так как данные идентифицируются по именам атрибутов и извлекаются независимо от порядка
   - *Атомарность значений* - каждая ячейка в таблице содержит одно неделимое значение
   - *Однородность доменов* - значения атрибутов берутся из одного и того же домена (типа данных), чтобы сохранить единообразие
   - *Целостность данных* - значения атрибутов должны соответствовать определенным правилам и ограничениям, например, `NOT NULL` для обязательных полей или ограничение уникальности для первичного ключа
5. ***Оператор `SELECT` и порядок выполнения операторов:***
   `SELECT` - оператор, который используется для извлечения данных из одной или нескольких таблиц, он позволяет указать, какие поля вернуть, какие условия применить и в каком порядке вывести данные

   Порядок выполнения операторов:
   1. `FROM`
   2. `JOIN`
   3. `WHERE`
   4. `GROUP BY`
   5. `HAVING`
   6. `SELECT`
   7. `DISTINCT`
   8. `ORDER BY`
   9. `LIMIT`
6. ***Определение База данных:***
   База данных - набор постоянно хранимых данных, используемых прикладными системами какого-либо предприятия

7. ***Уровни архитектуры ANSII/Sparc:***
   ANSII/Sparc - это стандарты подходов к построению архитектуры БД:
   1. *Внутренний уровень* - отвечает за физическое хранение данных на диске и оптимизацию работы с ними. Скрыт от пользователя, влияет на производительность и безопасность
   2. *Концептуальный уровень* - логическая структура данных (сущности, атрибуты, связи), с которой работают разработчики при проектировании базы
   3. *Внешний уровень* - представление данных для пользователей, показывающее нужную информацию и скрывающее технические детали хранения
8. ***Определение тета-соединения:***
   Тета-соединение — определяет отношение, содержащее кортежи из декартового произведения `R x S`, удовлетворяющие предикату `F = R.a_i θ S.b_j`, где `θ` — одна из операций сравнения `(>, <, =, ...)`.
9. ***Определение операции "проекция":***
   Проекция `П_{a1, ..., an}(R)` — результатом является новое отношение, содержащее вертикальное подмножество исходного отношения, создаваемое посредством извлечения указанных атрибутов и исключения из результата дублирующихся атрибутов.
10. ***В чём отличие дублирования от избыточности?***
    Отличие заключается в том, что *избыточность* подразумевает, что есть возможность восстановить или получить данные на основе других записей, но при это они все равно хранятся повторно, хотя их можно было бы извлечь из других данных. В случае дублирования такой возможности нет - это просто повторение данных, которые никак не зависят от других записей не могут быть восстановлены из них.

    Дублирование — это прямое повторение одних и тех же данных в разных местах (например, хранение имени студента в нескольких таблицах), что может привести к несоответствиям при обновлении

    Избыточность — более широкое понятие, которое включает дублирование, но также охватывает излишние данные, которые не обязательны для уникальной идентификации объекта

Пример избыточности:
Таблица "Студенты"

ФИО	Номер группы	Факультет
Иванов И.И.	M3210	ФИТИП
Петров П.П.	M3211	ФИТИП
Сидоров С.С.	M3212	ФИТИП

Таблица "Группы"

Номер группы	Факультет
М3210	ФИТИП
М3211	ФИТИП
М3212	ФИТИП

В этом случае информация о факультете хранится как отдельный атрибут у каждого студента, хотя её можно получить, исходя из номера группы

Пример дублирования:
Таблица "Студенты" с дублированием

ФИО	Номер группы	Факультет	Форма обучения
Иванов И.И.	М3210	ФИТИП	Контракт
Петров П.П.	М3211	ФИТИП	Бюджет
Сидоров С.С.	М3212	ФИТИП	Контракт

В этом случае информация о факультете повторяется для каждого студента, принадлежащего одной и той же группе, но её нельзя восстановить из других данных — это просто дублирование значений, так как факультет зависит от группы, а не от каждого студента

11. ***Функциональная зависимость:***
    Функциональная зависимость между атрибутами - это связь, при которой каждому значению атрибута *X* соответствует ровно одно значение атрибута *Y*.

$$R : X \to Y$$

12. ***Когда отношение в 1-ой Нормальной Форме:***
    1-я Нормальная Форма - таблица находится в **первой нормальной форме**, если все ее атрибуты являются атомарными, то есть неделимыми
13. ***Степень отношения:***
    Степень отношения - это количество его атрибутов. Степень отношения показывает, сколько различных характеристик описывает объекты, которые есть в таблице

Вариант 2:

1. ***Что описывается на верхнем уровне архитектуры ANSI/SPARC?***
   Внешний уровень - представление данных для пользователей, показывающее

нужную информацию и скрывающее технические детали хранения. Этот уровень абстракции про то, как принимать эффективные решения на основе данных

2. ***Определение СУБД:***
   СУБД - программное обеспечение с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а так же осуществлять к ней контролируемый доступ
3. ***Типы связей в модели сущность связь:***
   - Один-к-одному
   - Один-ко-многим
   - Многие-ко-многим
4. ***В чем отличие реляционной и постреляционной моделей данных?***
   - *Реляционная модель данных* требует, чтобы каждое поле таблицы содержало неделимое (атомарное) значение. Связи между таблицами задаются через общие атрибуты, и структура данных строго фиксирована
   - *Постреляционная модель данных* позволяет хранить сложные данные в одном поле, например, массивы или JSON, что удобно для записей с разными атрибутами. Это снижает разряженность таблиц, но усложняет проверку данных и может замедлять поиск
5. ***Что такое кардинальность отношения в реляционной модели данных?***
   Кардинальность отношения - это количество кортежей, которое содержит отношение
6. ***Дайте определение операции "Выборка":***

   $$\textbf{Выборка } \sigma_{\text{предикат}}(R) -$$

   Операция, результатом которой является отношение, которое содержит только те кортежи из исходного отношения, которые удовлетворяют заданному условию (предикату)
7. ***Дайте определение операции "Естественное соединение":***

   $$\textbf{Естественное соединение } R \bowtie S$$

   Операция, которая выполняет соединение по эквивалентности двух отношений по всем общим атрибутам, из результатов которого исключили по одному экземпляру каждого атрибута
8. ***Дайте определение аномалии модификации:***
   Изменение данных в одной записи может потребовать внесения изменений в другие записи

```
Пример: Если изменить название факультета для группы M3212, то необходимо
внести изменения для всех студентов этой группы. Если не внести их для
одного из студентов, данные окажутся несогласованными
```

9. ***Порядок команд в SQL для написании запросов и порядок их выполнения:***
   Порядок команд при написании SQL-запросов:

   1. **SELECT**
   2. **DISTINCT**
   3. **FROM**
   4. **JOIN**
   5. **WHERE**
   6. **GROUP BY**
   7. **HAVING**
   8. **ORDER BY**
   9. **LIMIT**

   Порядок их выполнения:
   1. **FROM**
   2. **JOIN**
   3. **WHERE**
   4. **GROUP BY**
   5. **HAVING**
   6. **SELECT**
   7. **DISTINCT**
   8. **ORDER BY**
   9. **LIMIT**
10. ***Какие условия должны быть, чтобы отношение находилось во второй нормальной форме?***
    2-я Нормальная Форма - таблица находится во **второй нормальной форме**, если находится в первой нормальной форме и все неключевые атрибуты функционально полностью зависят от первичного ключа
11. ***Суперключ:***
    Суперключ - это атрибут (множество атрибутов), который единственным, уникальным образом идентифицирует кортеж.
12. ***Частичная функциональная зависимость:***
    Частичная функциональная зависимость - зависимость от части составного ключа
13. **Какие есть группы основных требований к информационным системам и какое между ними противоречие?**
    - Надежность и безопасность
    - Производительность и малое потребление памяти
    - Гибкость и расширяемость
      Увеличение надежности и безопасности может снижать производительность, а добавление гибкости — увеличивать сложность и снижать скорость работы

Вариант 3:

1. ***Определение "Данные":***
   - *Данные* - это отдельные факты или значения без какого-либо контекста, которые сами по себе не имеют смысла (из инета)
   - *Данные* — это поддающиеся многократной интерпретации представления информации в формализованном виде, пригодном для передачи, интерпретации и обработки (Маятин)
2. ***Концептуальный уровень ANSII/SPARC:***
   Концептуальный уровень - логическая структура данных (сущности, атрибуты, связи), с которой работают разработчики при проектировании базы
3. ***В чем отличие между концептуальной и логической модели данных?***
   Концептуальная модель данных отражает общее представление данных без привязки к техническим деталям, описывая основные сущности, их атрибуты и связи. Она фокусируется на том, какие данные важны и как они соотносятся

   Логическая модель данных конкретизирует концептуальную модель, добавляя структуру для реализации в СУБД. Она включает типы данных, ключи и ограничения, но остаётся независимой от конкретных технологий хранения данных
4. ***Определение База данных:***
   База данных - набор постоянно хранимых данных, используемых прикладными системами какого-либо предприятия
5. ***Плюсы и минусы иерархической модели данных:***
   - **Плюсы**:
     - Удобна для восприятия человеком, так как соответствует реальным иерархиям
     - Простая структура, легко понимать и использовать для хранения иерархических данных
     - Быстро при транзакциях
   - **Минусы**:
     - Дублирование данных, что приводит к избыточности
     - Сложности в поддержании целостности данных и обновлении при изменении иерархии
     - Нельзя сделать связь "многие-ко-многим"
6. ***Определение "Домен":***
   Домен - это множество допустимых значений атрибутов
7. ***Определение "Потенциальный ключ":***
   Потенциальный ключ — это суперключ, который не содержит подмножества, также являющегося суперключом этого отношения
8. ***Операция объединения:***
   Объединение двух отношений R и S определяет новое отношение, которое

включает все кортежи, содержащиеся только в R, все кортежи, содержащиеся только в S и кортежи, содержащиеся и в R, и в S с исключением дубликатов

9. ***Что такое левое-внешнее соединение в реляционной алгебре?***
   Левое внешнее соединение R ⟕ S - это естественное соединение, при котором в результирующее отношение включаются также кортежи отношения R, не имеющие совпадающих значений в общих атрибутах отношения S
10. ***Алгоритм соединения таблиц вложенными циклами:***

```
type Row struct {
        Attr int
        Data string
}

func nestedLoopJoin(R, S []Row) []Row {
        var F []Row

        for _, r := range R {
                for _, s := range S {
                        if r.Attr == s.Attr {
                                joinedRow := Row{Attr: r.Attr, Data:
r.Data + " " + s.Data}
                                F = append(F, joinedRow)
                        }
                }
        }

        return F
}
```

11. ***Определение "Аномалия удаления":***
    Аномалия удаления - удаление одной записи может случайно удалить важную информацию
12. ***Что такое полная функциональная зависимость?***
    Полная функциональная зависимость - явление, когда неключевой атрибут зависит от всего составного ключа
13. ***Когда отношение находится в 3-ей нормальной форме?***
    Таблица находится в *третьей нормальной форме*, если она находится во второй нормальной форме и все неключевые атрибуты независимы друг от друга и функционально полностью зависят от первичного ключа (отсутствуют транзитивные зависимости)

Вариант 4:

1. ***Определение "СУБД"***
   СУБД - программное обеспечение с помощью которого пользователи могут

определять, создавать и поддерживать базу данных, а так же осуществлять к ней контролируемый доступ

2. ***Внешний/внутренний ключ, супер ключ:***
   - *Внешний ключ* - это атрибут (множество атрибутов) внутри отношения, который соответствует потенциальному ключу некоторого (возможно того же самого) отношения
   - *Внутренний ключ* - атрибут, предназначенный для внутреннего использования в системе, например, технический идентификатор, который не связан напрямую со значением данных **(!!!Не уверен, правильно ли это!!!)**
   - *Супер ключ* - это атрибут (множество атрибутов), который единственным образом идентифицирует кортеж
3. ***Что такое полусоединение?***
   - *Разность R-S* - состоит из кортежей, которые есть в R, но отсутствуют в S. Разность двух отношений определена только если они совместны по объединению
   - *Полусоединение R ▷F S* - отношение, содержащее кортежи R, которые входят в тета-соединение R и S
4. ***Определение "кортеж" в реляционных моделях:***
   Кортеж - отдельная строка в таблице
5. ***Что такое транзитивная функциональная зависимость?***
   Транзитивная функциональная зависимость - зависимость через промежуточный атрибут (Y зависит от X, X не зависит от Y, Z зависит от Y), то зависимость A -> C - транзитивная
6. ***Внутренний уровень ANSII/SPARC:***
   Внутренний уровень - отвечает за физическое хранение данных на диске и оптимизацию работы с ними. Скрыт от пользователя, влияет на производительность и безопасность
7. ***Что такое разность?***
   - *Разность R-S* - состоит из кортежей, которые есть в R, но отсутствуют в S. Разность двух отношений определена только если они совместны по объединению
8. ***Отличия логической и физической модели данных:***
   - **Логическая модель** описывает данные и связи между ними без указания, как они будут физически храниться
   - **Физическая модель** детализирует способы хранения и доступ к данным, учитывая технические аспекты
9. ***В чем плюсы и минусы хранения в файловой системе (файловом хранилище)?***
   - *Минусы:*
     - *Отсутствие структурированности* - данные хранятся в виде файлов без строгой структуры

        - *Ограниченные возможности поиска*
        - *Отсутствие транзакций и целостности данных* - невозможно гарантировать целостность данных и выполнять атомарные операции
        - *Отсутствие параллельной работы* - нет возможности одновременно работать нескольким пользователям с файлами из-за возможных ошибок
    - *Плюсы:*
        - *Простота реализации и использования* — легко организовать и управлять, особенно для небольших объемов данных
        - *Низкие затраты на хранение* — данные размещаются без дополнительного слоя управления, что снижает стоимость
        - *Гибкость форматов данных* — можно хранить разные типы данных, от текстовых до бинарных файлов
        - *Независимость от платформы* — файлы можно перемещать между системами без преобразований
10. ***Плюсы и минусы сетевой модели данных:***
    - **Плюсы сетевой модели данных:**
        - *Гибкость связей* — поддерживает многие-ко-многим, что позволяет создавать сложные связи между данными
        - *Эффективность доступа* — позволяет быстро находить и связывать данные, благодаря оптимизации связей и указателей
        - *Экономия на дублировании данных* — связи создаются напрямую, что снижает избыточность данных по сравнению с иерархической моделью
    - **Минусы сетевой модели данных:**
        - *Сложность управления* — управление и администрирование такой модели требуют сложной логики и специализированного программного обеспечения
        - *Трудность обновлений* — изменения структуры могут потребовать значительных изменений в приложениях
        - *Ограниченная поддержка* — разработка и поддержка сетевой модели были частично вытеснены реляционной моделью, что ограничивает её применение

4 вариация прошлогодней рубежки

Ссылка: https://aboba228.notion.site/c8fe49fb2496491ebb4cd1861bbbcbbd

Вариант 1:

1. ***В чем разница плотного и разреженного индекса?***
    - *Плотный индекс* — хранит указатель на каждую запись, обеспечивая быстрый поиск для уникальных значений

- *Разреженный индекс* — хранит указатель только на первый элемент блока, занимая меньше места, но может потребовать дополнительного поиска в блоке

2. ***Недостатки представлений:**
    1. *Ограниченность обновлений* — не все представления поддерживают изменение данных, особенно сложные запросы
    2. *Снижение производительности* — сложные представления могут замедлять запросы, так как данные извлекаются на лету
    3. *Структурные ограничения* — любые изменения в таблицах требуют изменений в представлениях
    4. *Материализованные представления* — требуют места для хранения и периодического обновления, что увеличивает нагрузку на систему
3. ***Транзакция:***
    - *Транзакция* - это последовательность, набор операций, которые выполняются как единое целое. Если хотя бы одна операция из набора не выполнится, все изменения отменяются. Это гарантирует целостность данных
4. ***Какие системы попадают в класс безопасности “D”?***
   Сюда относятся те системы, которые не удовлетворяют требованиям других классов (A, B, C, D)
5. ***Перечислите типы идентификаторов (типы аутентификации):***
   Аутентификация - это механизм проверки идентификатора (сопоставление пользователя с его идентифи-катором)
   Бывает три вида идентификаторов:
    1. То, что пользователь знает. Например, пароль, кодовое слово и т.п.
    2. То, чем пользователь владеет. Например, телефон, ключ–карта и т.п.
    3. То, чем является неотъемлемой частью пользователя (чаще всего это биометрия). Например, отпечаток пальца и т.п.
6. ***В чем разница горизонтального и вертикального фрагментирования в распределенной БД?***
    - *Горизонтальное (шардирование)* — делит таблицу на подмножества кортежей, распределяя их по узлам
    - *Вертикальное* — делит таблицу на подмножества атрибутов, храня их отдельно, что усложняет поддержание целостности
7. ***Перечислите виды прозрачности в распределенных СУБД:***
   Прозрачность в распределенных СУБД — это свойство системы, скрывающее от пользователя детали распределения данных для удобства и согласованности работы с ними.
    1. *Прозрачность фрагментации* - скрывает от пользователя, что данные разделены на фрагменты

2. *Прозрачность расположения фрагмента* - скрывает физическое местоположение фрагментов, позволяя обращаться к данным, не зная их расположения
3. *Прозрачность количества реплик* - скрывает количество копий данных, гарантируя доступ к данным независимо от числа их реплик
4. *Прозрачность контроля доступа* - скрывает механизмы безопасности, предоставляя доступ к данным только авторизованным пользователям

8. ***Перечислите предпосылки к возникновению NoSQL решений:***
   - *Увеличение объёма хранимых данных* — необходимость масштабируемых решений.
   - *Слабо структурированные данные* — гибкость для хранения данных без строгой схемы.
   - *Высокая взаимосвязность данных* — поддержка графов и сложных связей.
   - *Архитектура информационных систем* — распределённые и облачные системы с горизонтальным масштабированием
9. ***Опишите принцип хранения данных в базах “ключ-значение”:***
   Данные хранятся в виде пар *"ключ-значение"*, где **ключ** уникально идентифицирует элемент, а *значение* — это связанные с ключом данные. Каждая пара записывается как отдельный объект, и доступ к данным осуществляется по ключу - благодаря этому получаем быстрый поиск и добавление
10. ***Опишите принцип построения логической модели базы знаний:***

$$M = \langle T, P, A, B \rangle$$

   - *T* — множество базовых знаний (алфавит).
   - *P* — множество синтаксических правил.
   - *A* — аксиомы.
   - *B* — множество правил вывода (применяя их к аксиомам, можно получить новые синтаксические правила).

Вариант 2:

1. ***В чем разница первичного индекса и кластерного индекса:***
   Первичный индекс — это индекс, построенный по первичному ключу при условии, что исходный файл отсортирован по нему же
   Индекс кластеризации — это индекс, построенный по ключевому или неключевому полю при условии, что исходный файл отсортирован по нему же
2. ***В чем разница представления замены и материализованного представления?***
   Представление - это динамически сформированный результат одной или нескольких реляционных операций, сохраненный в виде нового отношения. Делятся на два типа

Материализованные - на самом деле дубликаты хранятся в памяти, которые были поучены при выполнении некоторого запроса
Представления замены - хранит только сам запрос, а не его результат. При каждом обращении этот запрос подставляется в основной запрос, и данные извлекаются заново

3. ***Свойства транзакций (грубо говоря ACID)***
    1. *Атомарность (Atomicity)* - транзакция выполняется полностью или не выполняется вовсе. Если одна часть операции не удается, все изменения откатываются
    2. *Согласованность (Consistency)* - после завершения транзакции данные должны оставаться в согласованном состоянии, соблюдая все правила и ограничения базы данных
    3. *Изолированность (Isolation)* - параллельные транзакции не должны влиять друг на друга. Результат одной транзакции виден другим только после её завершения
    4. *Долговечность (Durability)* - после завершения транзакции все изменения сохраняются и остаются неизменными даже при сбоях системы
4. ***Какие системы попадают в класс безопасности "С":***
Класс С (Дискреционный доступ) - система относится к классу С, если она имеет систему аутентификации/идентификации и дискреционное управление доступом
5. ***Что такое дискреционный (избирательный) контроль доступа?***
Дискреционный (избирательный) контроль доступа — это метод управления доступом, при котором права доступа к объектам определяются индивидуально для каждой пары "субъект-объект". В этой модели для каждого субъекта (например, пользователя или группы) можно настроить конкретные права на доступ к объекту (например, файлу или записи в БД) с указанием, какие действия разрешены (чтение, запись, удаление и т.д.)
6. ***В чем суть стратегии раздельного (фрагментного) размещения данных в распределенной БД?***
Стратегия раздельного (фрагментного) размещения данных в распределенной базе данных заключается в том, что база данных делится на отдельные, не пересекающиеся фрагменты, и каждый фрагмент хранится на своём узле. Такая схема позволяет эффективно распределять данные и снижать нагрузку на отдельные узлы
7. ***В чем различие гомогенной и гетерогенной распределенной БД?***
Гомогенная распределённая база данных — это система, в которой все узлы используют одну и ту же СУБД и, как правило, одну модель данных

    Гетерогенная распределённая база данных — это система, в которой узлы могут использовать разные СУБД и даже разные модели данных. Это позволяет интегрировать разнообразные источники данных, но усложняет управление и согласованность данных
8. ***Перечислите общие характеристики NoSQL решений:***

- *Отсутствие SQL* — используются альтернативные языки запросов, не основанные на SQL.
- *Отсутствие жесткой схемы (Schemaless)* — данные могут храниться без фиксированной структуры, что повышает гибкость.
- *Поддержка агрегатов (Aggregates)* — возможность хранить частично агрегированные данные для повышения производительности, хотя это может вызвать избыточность.
- *Слабые свойства ACID (BASE)* — NoSQL решения часто придерживаются модели BASE:
  - *Basic Availability* — каждый запрос завершается ответом, пусть даже временным.
  - *Soft State* — данные могут быть в изменяемом состоянии в любое время.
  - *Eventual Consistency* — система достигнет консистентного состояния после завершения всех операций.

9. ***Принцип хранения данных в документоориентированных БД:***
   В документоориентированных базах данные хранятся в виде самостоятельных документов (например, JSON, BSON, XML), содержащих вложенные структуры, массивы и пары «ключ-значение»
10. ***Принцип построения сетевой модели базы данных:***
    В сетевой модели данных информация представлена в виде графа, где сущности связаны отношениями типа "многие ко многим". В этой структуре узлы графа — это записи данных (например, объекты), а рёбра — связи между ними. Каждая запись может иметь несколько родителей и потомков