# 1. Для чего используются объектные типы данных в Oracle?

попа

– Расширяют реляционную модель

– Могут объединять данные и операции над ними

– Могут эффективно использоваться

– Могут показывать взаимосвязь и наследование данных

# 2. Кратко напишите создание типа в Oracle.

Тип в Oracle создается с помощью команды

**CREATE TYPE имя\_типа AS OBJECT (атрибут1 тип1, атрибут2 тип2, ...);.**

Тип может быть основан на стандартных типах данных или на других, ранее созданных, объектных типах.

# 3. Какие методы может содержать объектный тип в Oracle?

Объектный тип в Oracle может содержать методы member (принадлежащие экземпляру), методы-конструкторы, статические методы (принадлежащие типу) и методы сравнения (MAP или ORDER)

# 4. Какие таблицы могут хранить объекты в Oracle?

Объекты в Oracle могут храниться в:

1. Объектных таблицах: таблица, где каждая строка является экземпляром объектного типа (CREATE TABLE ... OF тип\_объекта;).
2. Таблицах, содержащих объекты: обычные таблицы, имеющие один или несколько столбцов объектного типа наряду с другими столбцами.

# 5. Как можно получить значение объекта из объектной таблицы?

Значение объекта из объектной таблицы можно получить с помощью функции VALUE(). Пример:

SELECT VALUE(псевдоним) FROM объектная\_таблица псевдоним;.

# 6. Как производятся DML операции с объектами?

DML операции (INSERT, UPDATE, DELETE) с объектами производятся аналогично стандартным таблицам.

1. При работе с таблицами, содержащими объекты в столбцах, для доступа к атрибутам объекта используется точечная нотация (псевдоним\_таблицы.имя\_столбца\_объекта.атрибут).
2. В объектных таблицах к атрибутам можно обращаться напрямую в условиях WHERE или SET.

# 7. Как осуществляется связь между объектами по аналогии со связями внешних ключей?

Вместо внешних ключей для связи между объектами используются ссылки (REF), которые основаны на идентификаторах объектов (OID). При создании таблицы столбец объявляется как REF имя\_типа SCOPE IS таблица\_объектов.

# 8. Как происходит наследование типов в Oracle?

Наследование типов позволяет создавать подтип на основе существующего супертипа с помощью ключевого слова UNDER (CREATE TYPE подтип UNDER супертип ...;). Супертип должен быть объявлен как NOT FINAL, чтобы от него можно было наследоваться.

# 9. Опишите методы сравнения для объектов Oracle.

Существует два типа методов сравнения (объект может иметь только один из них):

* MAP: Метод, который отображает объект на скалярное значение (например, число или строку). Это значение используется Oracle для сравнения объектов (>, <, =, сортировки, UNION).
* ORDER: Метод, который сравнивает два объекта (SELF и переданный параметр) и возвращает отрицательное число, ноль или положительное число, указывающее, меньше ли, равен ли или больше ли SELF переданного объекта. Используется для точного определения порядка.

# 10. Для чего применяются статические методы объектных типов Oracle?

Статические методы относятся к типу в целом, а не к конкретному экземпляру объекта. Они вызываются через имя типа (ИмяТипа.статический\_метод()) и используются для операций, не зависящих от состояния конкретного экземпляра, например, для утилитарных функций или операций над всеми объектами типа

STATIC FUNCTION from\_polar(radius NUMBER, angle\_radians NUMBER) RETURN complex\_number\_t,

# 11. Для чего применяются объектные представления?

Объектные представления используются для создания объектно-ориентированного интерфейса (аналога ORM) поверх существующих реляционных таблиц. Они позволяют работать с данными реляционных таблиц как с объектами.

# 12. Как реализовано индексирование для объектов Oracle?

Индексирование для объектов может быть реализовано по атрибутам объекта (используя точечную нотацию в CREATE INDEX) или по результату выполнения детерминированного метода объекта

# 13. Перечислите типы коллекций в Oracle.

В Oracle существуют три типа коллекций:

* Ассоциативные массивы (Associative arrays / Index-by tables)
* Вложенные таблицы (Nested tables)
* Массивы переменной длины (Varrays)

# 14. Как объявляется коллекция в Oracle?

Объявление коллекции происходит в два этапа:

1. Объявляется тип коллекции (в пакете, схеме (CREATE TYPE) или PL/SQL блоке):

TYPE имя\_типа\_коллекции IS TABLE OF тип\_элемента INDEX BY тип\_ключа; (для ассоциативных массивов)

TYPE имя\_типа\_коллекции IS TABLE OF тип\_элемента; (для вложенных таблиц)

TYPE имя\_типа\_коллекции IS VARRAY(размер) OF тип\_элемента; (для VARRAY)

1. Объявляется переменная этого типа: имя\_переменной имя\_типа\_коллекции;.

# 15. Из каких типов данных может состоять коллекция?

Коллекции состоят из однородных элементов, то есть все элементы в одной коллекции должны иметь один и тот же тип данных. Это может быть скалярный тип (NUMBER, VARCHAR2, DATE и т.д.), тип записи (RECORD) или объектный тип.

# 16. Что такое плотность и разреженность коллекции?

* Плотность (плотная коллекция): Все элементы между первым и последним индексом существуют (нет "пропусков" в индексах).
* Разреженность (разреженная коллекция): Существуют "пропуски" в последовательности индексов, то есть не для каждого индекса между первым и последним есть соответствующий элемент.

# 17. Какие коллекции всегда плотные?

Массивы переменной длины (Varrays) всегда являются плотными

# 18. Какие коллекции могут быть разреженными?

Разреженными могут быть:

* Вложенные таблицы (Nested tables) – становятся разреженными после удаления элементов не с конца.
* Ассоциативные массивы (Associative arrays) – могут быть разреженными изначально и поддерживают не последовательные индексы.

# 19. Что такое ограниченность (неограниченность) коллекции?

* Ограниченность: Коллекция имеет заранее определенный максимальный размер (верхнюю границу количества элементов или индексов).
* Неограниченность: Коллекция не имеет предопределенного максимального размера (хотя существуют практические ограничения).
* Коллекции типа VARRAY всегда ограничены.Вложенные таблицы и ассоциативные массивы неограничены.

# 20. Какие вам известны методы коллекций?

COUNT, DELETE, TRIM, EXTEND, FIRST, LAST, EXISTS, PRIOR, NEXT, LIMIT

**"С**егодня **Д**нём **Т**ы **Е**шь **Ф**рукты, **Л**имон, **Е**жевику, **П**отом **Н**ектарин, **Л**юбимый!"

# 21. Какие вам известны псевдофункции коллекций?

TABLE (преобразование коллекции в табличные данные), MULTISET (преобразование табличных данных в коллекцию), CAST (преобразование одного типа коллекции в другой)

# 22. Перечислите известные вам операции MULTISET для коллекций.

Операции MULTISET включают:

1. сравнение на равенство/неравенство (=, <>, !=),
2. MULTISET UNION [ALL | DISTINCT],
3. MULTISET INTERSECT [DISTINCT],
4. MULTISET EXCEPT [DISTINCT],
5. [NOT] MEMBER OF,
6. IS [NOT] EMPTY,
7. IS [NOT] A SET,
8. SUBMULTISET OF,
9. SET()

# 23. Перечислите, как могут использоваться коллекции.

Коллекции могут использоваться как: поле таблицы, параметр процедуры/функции, возвращаемое функцией значение, компонент записи, атрибут объектного типа

# 24. Для чего используется цикл FORALL при обработке коллекций?

Цикл FORALL используется для выполнения массовых (bulk) операций DML (INSERT, UPDATE, DELETE) на основе данных из коллекции. Он минимизирует переключение контекста между PL/SQL и SQL, что значительно повышает производительность по сравнению с поэлементной обработкой в обычном цикле

# 25. Для чего используется BULK COLLECT INTO при обработке коллекций?

BULK COLLECT INTO используется для выборки данных из SQL запроса (SELECT или DML с RETURNING) сразу в одну или несколько коллекций. Это также минимизирует переключение контекста и ускоряет получение больших объемов данных из БД в PL/SQL переменные

# 26. Что такое конвейерные функции в Oracle?

Конвейерная функция (pipelined function) - это вид табличной функции в PL/SQL, которая может возвращать строки результата вызывающей стороне по мере их генерации, а не накапливать весь результат в коллекции перед возвратом. Для этого используется ключевое слово PIPELINED в объявлении функции и команда PIPE ROW() для отправки каждой строки.

# 27. Для чего используются конвейерные функции в Oracle?

Конвейерные функции используются для:

* Возврата результатов (коллекции строк) из PL/SQL кода, которые могут быть обработаны как обычная таблица в SQL запросах (SELECT \* FROM TABLE(конвейерная\_функция(...))).
* Эффективной обработки и трансформации данных, особенно больших объемов, так как результат возвращается по частям, не требуя больших затрат памяти на хранение промежуточной коллекции.
* Параллельного выполнения.

# 28. Что такое пакет в Oracle?

Пакет (package) в Oracle - это объект схемы базы данных, который логически группирует связанные PL/SQL элементы. Пакет состоит из двух частей: спецификации (объявление публичных элементов) и тела (реализация процедур/функций и объявление приватных элементов).

# 29. Что может входить в пакет в Oracle?

В пакет могут входить:

* Процедуры и функции
* Переменные и константы
* Курсоры
* Исключения
* Типы данных (например, RECORD, коллекции, REF CURSOR)
* Секция инициализации (код, выполняемый при первом обращении к пакету в сессии)

# 30. Что такое постоянство пакетных переменных?

Постоянство (сохранение состояния) пакетных переменных зависит от того, объявлен ли пакет с PRAGMA SERIALLY\_REUSABLE:

* По умолчанию (без прагмы): Состояние публичных и приватных переменных пакета сохраняется на протяжении всей сессии пользователя. Секция инициализации выполняется один раз за сессию при первом обращении к пакету.
* С : Состояние переменных сохраняется только на время одного вызова серверу (например, одного вызова процедуры). При следующем вызове пакет "переинициализируется". Это требует меньше памяти в SGA, но состояние не сохраняется между вызовами.

# 31. Поясните использование пакетных исключений.

Пакет позволяет объявлять пользовательские исключения в своей спецификации. Эти исключения становятся доступны для использования как внутри тела пакета (для RAISE), так и вне пакета (для WHEN ... THEN в блоке EXCEPTION). Исключениям можно присвоить пользовательский номер ошибки с помощью PRAGMA EXCEPTION\_INIT

# 32. Поясните использование пакетных курсоров.

Курсоры могут быть объявлены в спецификации пакета, делая их доступными для внешнего использования. Это инкапсулирует SQL-запрос внутри пакета. Пакетные курсоры могут быть параметризованными. Внешний код может использовать их в неявных циклах (FOR record IN package.cursor\_name(...) LOOP) или явно управлять ими (OPEN package.cursor\_name(...); FETCH ...; CLOSE ...;)

# 33. Что такое сборки в SQL Server?

Сборки (assemblies) в SQL Server — это скомпилированные блоки кода (.NET DLL), написанные на языках CLR (например, C# или VB.NET), которые регистрируются внутри базы данных SQL Server с помощью команды CREATE ASSEMBLY. Они содержат реализацию CLR-объектов базы данных

# 34. Для чего используются сборки SQL Server?

Сборки используются для реализации логики, которую сложно или невозможно выполнить средствами T-SQL, например: сложная математика, обработка строк, доступ к внешним ресурсам (файлы, сеть, веб-службы), реализация пользовательских типов и агрегатов

# 35. Что может входить в сборки SQL Server?

Сборки могут содержать код для следующих объектов SQL Server: CLR хранимые процедуры, CLR функции (скалярные и табличные), CLR триггеры, CLR пользовательские типы (UDT), CLR пользовательские агрегаты (UDA)

# 36. Как сборка используется в SQL Server? ?????????????????????

Сначала разработчик **пишет код** на одном из языков .NET, поддерживаемых CLR. Этот код будет содержать логику, которую планируется выполнять внутри SQL Server.

Затем этот **код компилируетс**я с помощью .NET компилятора (например, csc.exe для C#) в файл сборки (обычно с расширением .dll).

**Включение интеграции с CLR** (если еще не включено):

По умолчанию возможность выполнения CLR-кода в SQL Server может быть отключена. Ее необходимо включить на уровне экземпляра сервера:

sp\_configure 'clr enabled', 1;

RECONFIGURE;

Скомпилированная DLL-ка загружается (регистрируется) в конкретную базу данных SQL Server с помощью команды CREATE ASSEMBLY.

CREATE ASSEMBLY MyCustomAssembly

FROM 'C:\Path\To\MyCustomAssembly.dll' -- Путь к файлу сборки

WITH PERMISSION\_SET = SAFE; -- Уровень разрешений (SAFE, EXTERNAL\_ACCESS, UNSAFE)

PERMISSION\_SET определяет, какие ресурсы доступны коду сборки:

* SAFE: Самый строгий. Код не может получить доступ к внешним ресурсам (файлы, сеть, реестр).
* EXTERNAL\_ACCESS: Позволяет доступ к некоторым внешним ресурсам (файлы, сеть), но с ограничениями безопасности SQL Server.
* UNSAFE: Позволяет практически неограниченный доступ, включая вызов неуправляемого кода. Требует максимальной осторожности.

После регистрации сборки ее публичные статические методы можно "опубликовать" в SQL Server как обычные SQL-объекты (хранимые процедуры, функции, триггеры и т.д.).

Это делается с помощью команд CREATE PROCEDURE, CREATE FUNCTION, CREATE TRIGGER, CREATE TYPE, CREATE AGGREGATE с указанием AS EXTERNAL NAME.

Синтаксис EXTERNAL NAME: ИмяСборки.[ПространствоИмен.ИмяКласса].ИмяМетода

После создания этих "оберток" SQL-объекты, основанные на CLR-сборке, используются в T-SQL коде точно так же, как и любые "родные" объекты SQL Server.

Когда SQL Server выполняет такой объект, он загружает соответствующую сборку, находит нужный метод и выполняет его код в среде CLR, интегрированной в процесс SQL Server.

**Короткая версия от анастасии в:**

1. EXEC sp\_configure 'clr enabled', 1; RECONFIGURE;

2. Сначала .NET код компилируется в сборку (DLL-файл). Эта DLL-сборка затем регистрируется в базу данных SQL Server с помощью команды CREATE ASSEMBLY.

3. После загрузки сборки, методы внутри этой сборки связываются с объектами SQL Server (объекты создаются с указанием EXTERNAL NAME).

4. После создания такого объекта SQL Server, он используется в T-SQL коде точно так же, как и любой обычный объект SQL Server.

# 37. Какие технологии входят в список технологий высокой доступности?

* RAC / RAC One: Обеспечивает отказоустойчивость на уровне сервера внутри площадки (Server HA) и масштабируемость.
* Flashback: Помогает быстро исправлять ошибки пользователя, минимизируя время простоя из-за логических сбоев.
* Application Continuity: Обеспечивает отказоустойчивость на уровне приложения (Application HA), скрывая сбои БД от конечного пользователя.
* ASM (Automatic Storage Management): Защита локального хранилища (зеркалирование), предотвращение простоев из-за сбоев дисков.
* Edition-based Redefinition, Online Redefinition: Технологии для обновлений приложений и структуры БД с минимальным или нулевым временем простоя (сокращение планируемых простоев).
* Active Data Guard: Обеспечивает защиту данных, аварийное восстановление (DR) при сбое основной площадки и возможность разгрузки основной БД (Query Offload).
* GoldenGate: Используется для репликации (в том числе active-active) и миграций с минимальным простоем, может работать с гетерогенными системами.
* Global Data Services: Обеспечивают автоматическое переключение сервисов (Service Failover) между площадками и балансировку нагрузки.
* RMAN, Oracle Secure Backup, Zero Data Loss Recovery Appliance: Инструменты для резервного копирования и быстрого восстановления после сбоев.
* Enterprise Manager Cloud Control: Координированное управление переключением между площадками (Coordinated Site Failover).
* Sharding: Основная цель - масштабируемость, изоляция сбоев в рамках одного шарда может повысить общую доступность системы.

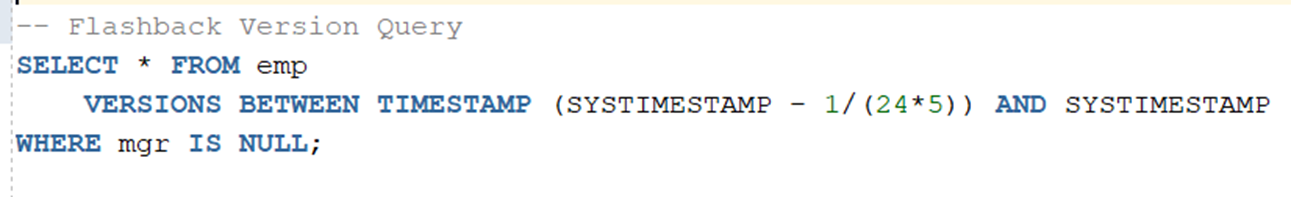
# 38. Что такое **ретроспективные запросы**?

Ретроспективные запросы позволяют увидеть данные, какими они были в некоторый момент в прошлом

# 39. Какие виды ретроспективных запросов вы знаете?

* Ретроспективные запросы – Flashback Queries( Позволяют увидеть данные таблицы (или таблиц) по состоянию на определенный момент в прошлом.)
* Ретроспективные версии данных – Flashback Version Query – возможность просмотра изменений данных в интервале времени
* Ретроспективные транзакции – Flashback Transaction Query – просмотр изменений, сделанных определенной транзакцией
* Технология Flashback Data Archive или Total Recall позволяет сохранять информацию об изменениях вне табличного пространства UNDO

# 40. Как получить ретроспективные версии данных?



SELECT \* FROM your\_table

VERSIONS BETWEEN { TIMESTAMP | SCN } start\_point AND { TIMESTAMP | SCN } end\_point

WHERE <условия\_фильтрации\_строк>; -- Опционально

# 41. Что такое **репликация**?

Репликация – механизм синхронизации нескольких копий объекта. Позволяет распределять данные из одной базы данных в одну или несколько других.

# 42. Когда применяется репликация?

* Синхронизация изменений удаленных баз данных с центральной БД
* Создание нескольких экземпляров БД, позволяющих распределить рабочую нагрузку
* Перемещение определенных наборов данных с центрального сервера распределение их на другие
* Преобразование данных и распределение их среди подписчиков

# 43. Когда не требуется применять репликацию?

* Однократно создать копию БД – копирование БД
* Скопировать данные с одного сервера на другой с преобразованием – процедура импорта-экспорта

# 44. Перечислите основные «действующие лица» репликации?

**Издатель** (Publisher):Издатели - серверы, предоставляющие данные для репликации на другие серверы.

**Дистрибьютор**(Distributor): Серверы, распространяющие реплицируемые данные.

**Подписчик** (Subscriber): Серверы назначения, которые хранят реплицированные данные и получают обновления.

**Публикация** (Publication): это совокупность статей, предназначенных для подписчиков.

**Статья** (Article): Статьи являются базовыми единицами, публикации

**Агенты репликации** (Replication Agents): Специальные программы, выполняющие задачи репликации .

# 45. Что такое топология репликации?

Топология репликации — это схема или архитектура, основная задача которой определить физическое расположение баз данных издателя, дистрибьютора и подписчика.

# 46. Какие бывают виды репликации?

* Репликация моментальных снимков (Snapshot Replication)
* Репликация сведением (слиянием, объединением) (Merge Replication)
* Транзакционная репликация (репликация транзакций) (Transact Replication)

# 47. Перечислите основные действия, необходимые для проведения репликации.

1. Выбор топологии репликации
2. Выбор типа репликации
3. Настройка дистрибьютора
4. Создание баз данных публикаций
5. Настройка издателя
6. Создание публикации
7. Создание подписок на издателе
8. Создание подписок на публикацию и назначение подписчиков
9. Проверка работы

# 48. Какие агенты могут использоваться для контроля репликации?

* Snapshot Agent (snapshot.exe)
* Distribution Agent (distrib.exe)
* Merge Agent (replmerg.exe)
* Log Reader Agent (logread.exe)
* Queue Reader Agent (qrdrsvc.exe)

# 49. Что такое **хранилище данных**?

Хранилище данных – база данных всех данных организации, к которым пользователи могут иметь единообразный доступ

# 50. Что такое консолидация данных?

Консолидация данных – все одинаковые запросы, выполняемые к хранилищу данных в разное время, должны возвращать одинаковые результаты

# 51. Что такое ETL?

Процесс консолидации – ETL (Extract, Transform, Load)

* сбор данных из различных источников
* очистка данных
* обеспечение качества данных

# 52. Что такое пространственная модель данных?

Многомерная модель данных — это логическая структура данных, оптимизированная для запросов и анализа в хранилищах данных и системах OLAP (Online Analytical Processing). ?????????????

# 53. Что хранится в таблице фактов?

Содержит большинство строк хранилища данных

Имеет составной первичный ключ

Неключевые столбцы имеют числовой тип данных

Основная задача – дать сумму, максимальное, минимальное, среднее и пр. значение

# 54. Что хранится в таблице измерений?

Содержит сравнительно небольшой набор строк

Первичный ключ состоит из одного столбца

Столбцы содержат описания измерений

Разреженность измерений

# 55. Что такое агрегирование данных?

Агрегатные таблицы – aggregate tables – промежуточные таблицы для хранения заранее подсчитанных низкоуровневых данных

Процесс создания — агрегирование данных

# 56. Что такое **куб**?

Куб — подмножество данных из хранилища данных, которое можно организовать в многомерную структуру

# 57. Что такое измерение?

Измерение – описание для элемента данных, по которому проводится анализ

# 58. Что такое мера?

Мера – значения в ячейке.

Ячейка – набор точек пространства многомерного куба

# 59. Перечислите этапы создания куба.

* Создание проекта бизнес-аналитики
* Определение источников данных
* Создание представлений источников данных
* Создание куба (таблица фактов, измерения)
* Агрегирование
* Обработка куба
* Просмотр куба

# 60. Какой продукт использовался при выполнении лабораторных работ по созданию куба?

SQL Server Data Tools

# 61. Перечислите основные операторы языка MDX.

MDX – язык запросов для извлечения данных из многомерных БД

* SELECT
* FROM
* WHERE
* Комментарии -- // /\* …\*/
* Арифметические операторы + - \* /
* Операторы для наборов + - \*
* Логические операторы > < <= >= = <> AND OR XOR NOT
* Специальные операторы {} , :

# 62. Поясните использование Power Pivot при работе с кубом.

Power Pivot — это технология, которая позволяет выполнять мощный анализ данных.

* Пользователь в Excel может подключиться к внешнему источнику данных — серверу Analysis Services. Указывается имя сервера SSAS и выбирается нужный куб.
* После подключения создается сводная таблица , подключенная к кубу SSAS.
* Пользователь работает с полями сводной таблицы, которые отражают структуру куба.
* Перетаскивая элементы из списка полей в области сводной таблицы , пользователь формирует нужный срез данных.
* Можно использовать срезы (Slicers) и временные шкалы (Timelines) Excel для интерактивной фильтрации данных куба.

# 63. Что такое **NoSQL** решения?

NoSQL – набор различных решений для построения гибких БД. Оптимизированы для работы с большим объемом данных за счет отсутствия контроля целостности

# 64. Какие преимущества и недостатки NoSQL решения?

Преимущества

* Гибкость
* Быстрее разработка
* Возможна поэтапная реализация
* Частично структурированные или неструктурированные данных
* Масштабируемость
* Использование кластеров
* Высокая производительность
* Оптимизированы для конкретных шаблонов
* Широкие функциональные возможности

Недостатки

* Слабая согласованность (Eventual Consistency)
* Ограниченные запросы / JOIN'ы
* Сложные / Ограниченные транзакции (ACID)
* Сложность моделирования (для эффективности)

# 65. Какие виды NoSQL решений вы знаете?

* Ключ‑значение
* Документные базы данных
* Графовые базы данных
* InMemory базы данных
* Поисковые базы данных

# 66. Что такое **MongoDB**?

Документо-ориентированная система управления базами данных

* База данных представляет собой набор коллекций
* Коллекции содержат документы

# 67. Что такое коллекция MongoDB?

Коллекция (Collection) в MongoDB — это группа документов. Она является аналогом таблицы в реляционных базах данных.

# 68. Что такое документ MongoDB?

Документ (Document) — это основная единица данных в MongoDB. Он представляет собой структуру данных, состоящую из пар "поле": "значение", аналогичную объекту JSON. Для каждого документа имеется уникальный идентификатор, который называется \_id

# 69. Как устанавливается сервер MongoDB?

* Перейти на официальный сайт MongoDB (mongodb.com).
* Выбрать нужную версию и тип пакета для вашей операционной системы
* Запустить скачанный MSI-инсталлятор. Следовать указаниям мастера установки.
* Запустить mongod
* Подключиться к серверу с помощью клиента

# 70. Какой клиент использовался в лабораторных работах по MongoDB?

Графический клиент Compass

# 71. Перечислите функции MongoDB, аналогичные DML операциям.

INSERT: **db.collection.insertOne(), db.collection.insertMany()**

UPDATE: **db.collection.updateOne(), db.collection.updateMany()**

DELETE: **db.collection.deleteOne(), db.collection.deleteMany()**

SELECT: **db.collection.find()**

# 72. Напишите аналог оператора DELETE.

**db.collection.deleteOne({ <условие> })** - удаляет один документ, соответствующий условию.

**db.collection.deleteMany({ <условие> })** - удаляет все документы, соответствующие условию.

# 73. Напишите аналог оператора UPDATE.

**db.collection.updateOne({ <условие> }, { $set: { <новые значения> } })** - обновляет один документ.

**db.collection.updateMany({ <условие> }, { $set: { <новые значения> } })** - обновляет множество документов.

# 74. Напишите аналог оператора SELECT с условием BETWEEN.

Для поля field между value1 и value2 (включительно):

**db.collection.find({ field: { $gte: value1, $lte: value2 } })**

(Используются операторы $gte - больше или равно, и $lte - меньше или равно)

# 75. Напишите аналог оператора SELECT с условием LIKE.

**db.collection.find({ field: { $regex: "шаблон" } })**

# 76. Что такое проекция в MongoDB?

Проекция – это вывод только нужных полей из документов коллекции при выборке.

# 77. Как ограничить выборку MongoDB?

С помощью метода **.limit(N)**, где N – максимальное количество возвращаемых документов.

# 78. Как отсортировать выборку в MongoDB?

С помощью метода **.sort({ field: 1 })** для сортировки по возрастанию или **.sort({ field: -1 })** для сортировки по убыванию

# 79. Какие виды индексов есть в MongoDB?

* Простые индексы (на один атрибут)
* Составные индексы (на несколько атрибутов)
* Уникальные индексы
* Индексы покрытия
* Частичные индексы
* Пространственные индексы (2dsphere, 2d)
* Полнотекстовые индексы (text)

# 80. Как просмотреть индексы в MongoDB?

Через MongoDB Compass на вкладке "**Indexes**" для коллекции.

С помощью команды **db.collection.getIndexes()** в mongosh.

# 81. Как определить, сработал ли индекс MongoDB?

С помощью метода **.explain("executionStats")** для запроса. В выводе нужно смотреть на:

* **winningPlan.stage:** должен быть IXSCAN (сканирование индекса), а не COLLSCAN (сканирование коллекции).
* **totalDocsExamined:** должно быть равно или близко к nReturned (количество возвращенных документов), а не общему числу документов в коллекции.
* **executionTimeMillis**: время выполнения запроса

# 82. Какие виды пространственных индексов есть в MongoDB?

**2dsphere:** для данных, хранящихся в сферической геометрии (например, координаты на Земле).

**2d:** для данных на плоской (двумерной) модели.

# 83. Какие функции можно использовать с пространственными индексами в MongoDB?

**Для 2d индексов:**

* $geoWithin: нахождение объектов внутри фигуры ($box, $center, $polygon).
* $near: нахождение объектов поблизости.

**Для 2dsphere индексов:**

* $geoIntersects: пересечение объектов
* $near: нахождение объектов поблизости
* $geoWithin: нахождение объектов внутри фигуры ($geometry, $centerSphere)

# 84. Какие виды полнотекстовых индексов можно создавать в MongoDB?

MongoDB поддерживает один тип полнотекстового индекса (text), но его можно конфигурировать:

* С указанием конкретных полей для индексации.
* На все текстовые поля в коллекции (wildcard-индекс "$\*\*": "text").
* С указанием весов для различных полей (для влияния на релевантность).

# 85. Какие особенности использования полнотекстовых индексов в MongoDB вы знаете?

Поддерживают **поиск ИЛИ** (пробел между словами), **поиск И** (фразы в кавычках или экранированные кавычки).

* Поддерживают исключение слов (префикс -).
* Осуществляют стемминг (поиск по основе слова для разных форм одного слова).
* Рассчитывают релевантность (score) для каждого найденного документа.
* Используют токенизацию и стоп-слова.
* Одна коллекция может иметь не более одного полнотекстового индекса.

# 86. Что такое релевантность выборки?

**Релевантность выборки** (в контексте полнотекстового поиска) – это мера того, насколько найденный документ соответствует поисковому запросу. MongoDB вычисляет score (оценку релевантности), который учитывает частоту термина, веса полей, точность совпадения слова и количество слов в поиске.

# 87. Для чего используется фреймворк агрегации в MongoDB?

Фреймворк агрегации используется для формирования последовательной обработки значений из документов и выполнения сложных операций анализа данных, таких как группировка, вычисление итогов, преобразование структуры документов, объединение данных из разных коллекций и т.д.

# 88. Что такое пайплайн?

**Пайплайн** (pipeline) в MongoDB — это последовательность стадий (операций обработки данных), через которые проходят документы коллекции в рамках фреймворка агрегации.

# 89. Как создается пайплайн?

Пайплайн создается с помощью метода **db.collection.aggregate([])**, которому в качестве аргумента передается массив, состоящий из одной или нескольких стадий (документов, описывающих операции).

# 90. Как просмотреть и изменить пайплайн?

**Просмотреть**: В MongoDB Compass пайплайн можно увидеть в визуальном конструкторе на вкладке "**Aggregations**". В mongosh или коде — это массив стадий

**Изменить**: В MongoDB Compass — путем добавления, удаления или изменения стадий в визуальном конструкторе. В коде — путем модификации массива стадий.

# 91. Какие классы стадий пайплайна вы знаете?

* Query (Запрос): $match, $geoNear, $limit, $skip, $sort, $sample, $search, $searchMeta.
* Mutate (Изменение данных): $addFields, $densify, $fill, $project, $redact, $replaceRoot, $replaceWith, $set, $setWindowFields, $unset.
* Summarise/Itemise (Подсчет итогов/Детализация): $bucket, $bucketAuto, $count, $facet, $group, $sortByCount, $unwind.
* Join: $graphLookup, $lookup, $unionWith.
* Input/Output: $documents, $merge, $out.

# 92. Поясните, как запустить сервер MongoDB в режиме аутентификации?

Необходимо запустить сервер mongod с ключом --auth или добавить security.authorization: enabled в конфигурационный файл. Перед этим должны быть созданы пользователи.

Пример запуска из командной строки: mongod --auth

# 93. Что такое роль в MongoDB?

Роль в MongoDB — это набор привилегий, которые определяют, какие действия пользователь может выполнять и на каких ресурсах (например, чтение данных из определенной коллекции, администрирование базы данных).

# 94. Что такое пользователь MongoDB?

Пользователь MongoDB — это сущность, которая аутентифицируется в системе MongoDB и которой назначаются одна или несколько ролей для авторизации доступа к данным и выполнения операций

# 95. Есть ли встроенные роли в MongoDB?

Да, есть. Например: read, readWrite, dbAdmin, userAdmin, dbOwner, dbAdminAnyDatabase, root

# 96. Как назначить роль пользователю в MongoDB?

Роль назначается пользователю при его создании с помощью db.createUser() в параметре roles, либо существующему пользователю с помощью команд db.grantRolesToUser().

Пример при создании: db.createUser({ user: "myUser", pwd: "password", roles: [ { role: "readWrite", db: "myDatabase" } ] })

# 97. Какие технологии высокой доступности поддерживаются в MongoDB?

**Набор реплик** (Replica Set): Обеспечивает отказоустойчивость и избыточность данных путем хранения копий данных на нескольких серверах

**Шардинг** (Sharding): Обеспечивает горизонтальное масштабирование путем распределения данных по нескольким серверам или наборам репликы

# 98. Что представляет собой база данных MongoDB с точки зрения файлового хранения?

Каждая коллекция и каждый индекс обычно хранятся как отдельные файлы (или группы файлов, в зависимости от движка хранения, например, WiredTiger).

# 99. Что такое WiredTiger?

WiredTiger — это высокопроизводительный, масштабируемый движок хранения (storage engine), используемый в MongoDB по умолчанию. Он поддерживает построчное и столбцовое хранение, сжатие, LSM-деревья и транзакции. Ориентирован на запись.

# 100. Что такое LSM?

LSM (Log-structured merge-tree) — это журнально-структурированное дерево слияния, структура данных, оптимизированная для высокой пропускной способности записи. Записи сначала добавляются в Memtable (таблицу в памяти), затем, при достижении порога, записываются на диск как SSTable (отсортированная таблица строк). Периодически SSTable сливаются.

# 101. Что такое MMAP?

MMAP (Memory Mapped Storage Engine) — это более старый движок хранения MongoDB (до WiredTiger), который использует отображение файлов в память (memory-mapped files). Он ориентирован на чтение, имеет блокировку на уровне коллекции и может вызывать фрагментацию

# 102. В каком формате хранятся документы MongoDB?

Документы MongoDB хранятся в формате BSON (Binary JSON) — двоичном представлении JSON-подобных документов.

# 103. Какие типы данных поддерживает MongoDB?

String, Array, Binary data, Boolean, Date, JavaScript, Object, ObjectId, Integer (32-bit), Long (64-bit), Double, Decimal128, Regular expression.

# 104. Есть ли ограничения целостности в MongoDB?

Да, ограничения целостности (или валидация схемы) могут быть заданы для коллекций через механизм "Validation", который определяет правила, которым должны соответствовать документы при вставке или обновлении.

# 105. Как осуществляется резервное копирование в MongoDB?

С помощью набора утилит MongoDB Database Tools:

* mongodump – для создания полного дампа данных (бэкапа).
* mongorestore – для восстановления из дампа.
* bsondump – для конвертации BSON файлов в читаемый JSON формат.
* mongoexport / mongoimport – для экспорта/импорта данных в/из JSON или CSV.