Ответы на вопросы по базам данных

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

Объектные типы данных в Oracle позволяют определять и использовать пользовательские сложные структуры данных как единые объекты.

# 2. Кратко напишите создание типа в Oracle.

Тип в Oracle создается командой `CREATE TYPE имя\_типа AS OBJECT (атрибут1 тип\_данных, ...);`.

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

Объектный тип в Oracle может содержать MEMBER, STATIC, CONSTRUCTOR, MAP и ORDER методы.

# 4. Какие таблицы могут хранить объекты в Oracle?

Объекты в Oracle могут храниться в объектных таблицах, реляционных таблицах с объектными колонками, вложенных таблицах и таблицах с REF колонками.

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

Значение объекта из объектной таблицы можно получить через SELECT с указанием атрибутов, либо используя функции VALUE(), REF() или DEREF().

# 6. Как производятся DML операции с объектами?

DML операции (INSERT, UPDATE, DELETE, SELECT) с объектами производятся аналогично обычным таблицам, используя стандартный SQL или специальные объектные функции.

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

Связь между объектами в Oracle осуществляется через REF типы, вложенные объекты, коллекции или ассоциативные массивы.

# 8. Как происходит наследование типов в Oracle?

Наследование типов в Oracle происходит созданием подтипа с ключевым словом UNDER, который наследует атрибуты и методы родителя.

# 9. Опишите методы сравнения для объектов Oracle.

Для сравнения объектов Oracle используются MAP или ORDER методы, стандартные операторы сравнения и функция VALUE().

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

Статические методы объектных типов Oracle применяются для операций, не требующих экземпляра объекта, например, для фабричных методов или утилитарных функций.

# 11. Для чего применяются объектные представления?

Объектные представления в Oracle применяются для преобразования реляционных данных в объектную модель или инкапсуляции сложной логики доступа.

# 12. Как реализовано индексирование для объектов Oracle?

Индексирование для объектов Oracle реализовано через индексы по атрибутам, функциональные индексы для методов, индексы по REF типам, а также пространственные и текстовые индексы.

# 13. Перечислите типы коллекций в Oracle.

В Oracle существуют три типа коллекций: VARRAY (массивы), Nested Table (вложенные таблицы) и Associative Array (ассоциативные массивы).

# 14. Как объявляется коллекция в Oracle?

Коллекция VARRAY объявляется как `CREATE TYPE имя AS VARRAY(размер) OF тип;`, Nested Table как `CREATE TYPE имя AS TABLE OF тип;`, а Associative Array в PL/SQL как `TYPE имя IS TABLE OF тип INDEX BY тип\_индекса;`.

# 15. Из каких типов данных может состоять коллекция?

Коллекция в Oracle может состоять из скалярных типов, объектных типов, других коллекций, REF типов или LOB типов.

# 16. Что такое плотность и разреженность коллекции?

Плотная коллекция имеет последовательно пронумерованные элементы без пропусков, а разреженная коллекция может иметь пропуски в индексах.

# 17. Какие коллекции всегда плотные?

Всегда плотными коллекциями в Oracle являются VARRAY и Nested Table при использовании в таблицах.

# 18. Какие коллекции могут быть разреженными?

Разреженными коллекциями в Oracle могут быть Associative Array и Nested Table при использовании в PL/SQL.

# 19. Что такое ограниченность (неограниченность) коллекции?

Ограниченная коллекция (например, VARRAY) имеет фиксированный максимальный размер, в то время как неограниченная (Nested Table, Associative Array) может расти динамически.

# 20. Какие вам известны методы коллекций?

Основные методы коллекций в Oracle включают COUNT, DELETE, TRIM, EXTEND, FIRST/LAST, NEXT/PRIOR и EXISTS.

# 21. Какие вам известны псевдофункции коллекций?

Псевдофункции коллекций в Oracle включают CARDINALITY, SET, MULTISET, MULTISET UNION, MULTISET INTERSECT и MULTISET EXCEPT.

# 22. Перечислите известные вам операции MULTISET для коллекций.

Операции MULTISET для коллекций включают MULTISET UNION, INTERSECT, EXCEPT, CAST и SUBMULTISET.

# 23. Перечислите, как могут использоваться коллекции.

Коллекции в Oracle могут использоваться как колонки таблиц, параметры и возвращаемые значения процедур/функций, в PL/SQL блоках, как атрибуты объектных типов и в SQL запросах.

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

Цикл FORALL в Oracle используется для массовой DML-обработки элементов коллекции с целью повышения производительности.

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

BULK COLLECT INTO в Oracle используется для массовой выборки данных из курсора в коллекцию для повышения производительности.

# 26. Что такое конвейерные функции в Oracle?

Конвейерные функции в Oracle возвращают результаты построчно (через `PIPELINED`), позволяя обрабатывать данные потоково.

# 27. Для чего используются конвейерные функции в Oracle?

Конвейерные функции в Oracle используются для потоковой обработки, преобразования данных в реальном времени и создания виртуальных таблиц.

# 28. Что такое пакет в Oracle?

Пакет в Oracle — это именованная логическая группа связанных PL/SQL типов, переменных, констант, процедур, функций и курсоров.

# 29. Что может входить в пакет в Oracle?

В пакет Oracle могут входить процедуры, функции, переменные, константы, типы данных, коллекции, курсоры, исключения и даже другие пакеты.

# 30. Что такое постоянство пакетных переменных?

Постоянство пакетных переменных означает, что их значения сохраняются в течение сессии между вызовами процедур и функций пакета.

# 31. Поясните использование пакетных исключений.

Пакетные исключения, объявленные в спецификации пакета, позволяют стандартизировать обработку ошибок во всех его процедурах и функциях.

# 32. Поясните использование пакетных курсоров.

Пакетные курсоры, объявленные в спецификации, сохраняют свое состояние между вызовами и позволяют повторно использовать результаты запроса в рамках сессии.

# 33. Что такое сборки в SQL Server?

Сборки в SQL Server — это DLL-файлы, содержащие скомпилированный .NET код, используемый для расширения функциональности сервера.

# 34. Для чего используются сборки SQL Server?

Сборки SQL Server используются для создания пользовательских типов данных, хранимых процедур, функций, триггеров и агрегатов на .NET.

# 35. Что может входить в сборки SQL Server?

В сборки SQL Server могут входить пользовательские типы данных, хранимые процедуры, функции (скалярные, табличные), агрегатные функции и триггеры.

# 36. Как сборка используется в SQL Server?

Сборка используется в SQL Server после ее регистрации в базе данных (CREATE ASSEMBLY) путем создания и вызова объектов на ее основе.

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

Технологии высокой доступности включают репликацию, кластеризацию, зеркалирование, AlwaysOn Availability Groups, а также резервное копирование и восстановление.

# 38. Что такое ретроспективные запросы?

Ретроспективные запросы (или Flashback Query в Oracle) позволяют получать данные такими, какими они были на определенный момент времени в прошлом.

# 39. Какие виды ретроспективных запросов вы знаете?

Виды ретроспективных запросов включают запросы на определенный момент времени (AS OF TIMESTAMP) и запросы версий строк (VERSIONS BETWEEN).

# 40. Как получить ретроспективные версии данных?

Ретроспективные версии данных можно получить используя специальные SQL-конструкции, такие как `AS OF TIMESTAMP` или `VERSIONS BETWEEN` в Oracle.

# 41. Что такое репликация?

Репликация — это процесс копирования и синхронизации данных между несколькими базами данных для обеспечения доступности, распределения нагрузки или отказоустойчивости.

# 42. Когда применяется репликация?

Репликация применяется для обеспечения высокой доступности, распределения нагрузки, отказоустойчивости, географического распределения данных или синхронизации баз данных.

# 43. Когда не требуется применять репликацию?

Репликация не требуется, если данные не критичны, достаточно резервного копирования, нет требований к высокой доступности или распределению нагрузки.

# 44. Перечислите основные «действующие лица» репликации?

Основные «действующие лица» репликации: издатель (Publisher), распространитель (Distributor), подписчик (Subscriber), публикация (Publication) и статья (Article).

# 45. Что такое топология репликации?

Топология репликации определяет конфигурацию серверов (издателей, распространителей, подписчиков) и направление потоков данных между ними.

# 46. Какие бывают виды репликации?

Виды репликации включают транзакционную, репликацию слиянием, репликацию моментальных снимков и пиринговую (peer-to-peer) репликацию.

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

Основные действия для репликации: настройка издателя и распространителя, создание публикации и статей, настройка подписчиков и создание подписок.

# 48. Какие агенты могут использоваться для контроля репликации?

Для контроля репликации используются агенты: моментальных снимков, чтения журнала, распространителя, слияния и иногда агент очереди.

# 49. Что такое хранилище данных?

Хранилище данных (Data Warehouse) — это предметно-ориентированная, интегрированная, историческая и не изменяемая совокупность данных для поддержки принятия управленческих решений.

# 50. Что такое консолидация данных?

Консолидация данных — это процесс объединения и интеграции данных из различных источников в единую систему для обеспечения их согласованности.

# 51. Что такое ETL процесс?

ETL (Extract, Transform, Load) — это процесс извлечения данных из источников, их преобразования (очистка, стандартизация) и загрузки в целевую систему, например, в хранилище данных.

# 52. Что такое OLAP?

OLAP (Online Analytical Processing) — это технология многомерного анализа данных, позволяющая быстро выполнять сложные запросы к агрегированным данным для бизнес-аналитики.

# 53. Что такое OLTP?

OLTP (Online Transaction Processing) — это системы, предназначенные для обработки большого количества коротких транзакций в реальном времени, например, банковские операции.

# 54. Что такое NoSQL?

NoSQL — это класс нереляционных баз данных, оптимизированных для гибкой схемы, горизонтальной масштабируемости и работы с большими объемами разнообразных данных.

# 55. Какие типы NoSQL баз данных вы знаете?

Основные типы NoSQL баз данных: документоориентированные (MongoDB), ключ-значение (Redis), колоночные (Cassandra) и графовые (Neo4j).

# 56. Что такое MongoDB?

MongoDB — это документоориентированная NoSQL база данных, хранящая данные в гибких BSON-документах (бинарный JSON) и поддерживающая масштабируемость.

# 57. Как создается база данных в MongoDB?

База данных в MongoDB создается автоматически при первой вставке данных в коллекцию этой базы или командой `use имя\_базы\_данных`.

# 58. Как создается коллекция в MongoDB?

Коллекция в MongoDB создается автоматически при первой вставке документа в нее или явно командой `db.createCollection("имя\_коллекции")`.

# 59. Как вставляются документы в MongoDB?

Документы в MongoDB вставляются методами `insertOne()` (один документ) или `insertMany()` (несколько документов).

# 60. Как обновляются документы в MongoDB?

Документы в MongoDB обновляются методами `updateOne()`, `updateMany()` или `replaceOne()`, используя операторы обновления.

# 61. Как удаляются документы в MongoDB?

Документы в MongoDB удаляются методами `deleteOne()` (один документ по условию) или `deleteMany()` (несколько документов по условию).

# 62. Как выполняются запросы в MongoDB?

Запросы в MongoDB выполняются методом `find()` (для выборки нескольких документов) или `findOne()` (для одного документа), используя объект-фильтр.

# 63. Что такое агрегация в MongoDB?

Агрегация в MongoDB — это процесс обработки данных, включающий стадии группировки, фильтрации и преобразования для получения вычисляемых результатов.

# 64. Какие стадии агрегации вы знаете в MongoDB?

Основные стадии агрегации в MongoDB включают `$match`, `$group`, `$sort`, `$project`, `$limit`, `$skip`, `$unwind` и `$lookup`.

# 65. Что такое индексы в MongoDB?

Индексы в MongoDB — это специальные структуры данных, которые хранят небольшую часть набора данных коллекции в легкодоступной форме для ускорения запросов.

# 66. Какие типы индексов поддерживает MongoDB?

MongoDB поддерживает одиночные, составные, мультиключевые, геопространственные, текстовые, хешированные, TTL и частичные индексы.

# 67. Что такое репликация в MongoDB?

Репликация в MongoDB (replica set) — это процесс создания копий данных на нескольких серверах для обеспечения высокой доступности и отказоустойчивости.

# 68. Что такое шардирование в MongoDB?

Шардирование в MongoDB — это метод горизонтального масштабирования путем распределения данных коллекции по нескольким серверам (шардам).

# 69. Что такое GridFS в MongoDB?

GridFS в MongoDB — это спецификация для хранения и извлечения файлов, превышающих ограничение на размер BSON-документа (16МБ), путем их разделения на части.

# 70. Что такое MapReduce в MongoDB?

MapReduce в MongoDB — это парадигма обработки данных для выполнения агрегации по большим наборам данных, хотя сейчас предпочтительнее использовать Aggregation Pipeline.

# 71. Что такое валидация документов в MongoDB?

Валидация документов в MongoDB позволяет определить правила для структуры и типов данных полей, чтобы обеспечить их корректность при вставке или обновлении.

# 72. Что такое транзакции в MongoDB?

Транзакции в MongoDB позволяют выполнять последовательность операций чтения и записи на нескольких документах атомарно (как единое целое).

# 73. Что такое Change Streams в MongoDB?

Change Streams в MongoDB предоставляют API для приложений, чтобы в реальном времени отслеживать изменения данных в коллекциях, базах данных или всем развертывании.

# 74. Что такое TTL индексы в MongoDB?

TTL (Time-To-Live) индексы в MongoDB — это специальные однополевые индексы, которые автоматически удаляют документы из коллекции по истечении указанного времени.

# 75. Что такое частичные индексы в MongoDB?

Частичные индексы в MongoDB индексируют только те документы в коллекции, которые соответствуют указанному выражению фильтра, экономя место и повышая производительность.

# 76. Что такое покрывающие индексы в MongoDB?

Покрывающий индекс в MongoDB включает все поля, необходимые для выполнения запроса, позволяя получить результат напрямую из индекса без чтения документов.

# 77. Что такое сжатие в MongoDB?

Сжатие в MongoDB позволяет уменьшить размер хранимых данных на диске и объем передаваемых по сети данных, используя алгоритмы типа snappy, zlib или zstd.

# 78. Что такое кэширование в MongoDB?

MongoDB использует внутренний кэш (WiredTiger cache) для хранения в оперативной памяти наиболее часто используемых данных и индексов для ускорения доступа.

# 79. Что такое мониторинг в MongoDB?

Мониторинг в MongoDB включает сбор и анализ метрик производительности, состояния сервера и операций для диагностики проблем и оптимизации работы.

# 80. Что такое бэкап в MongoDB?

Бэкап в MongoDB — это процесс создания резервных копий данных для возможности их восстановления в случае сбоев или потери данных.

# 81. Что такое восстановление в MongoDB?

Восстановление в MongoDB — это процесс возврата данных из резервной копии или откат к определенной точке во времени после сбоя или ошибки.

# 82. Что такое миграция в MongoDB?

Миграция в MongoDB — это процесс переноса данных из одной среды MongoDB в другую, или из другой СУБД в MongoDB, часто с преобразованием схемы.

# 83. Что такое безопасность в MongoDB?

Безопасность в MongoDB включает аутентификацию, авторизацию (управление доступом на основе ролей), шифрование данных при хранении и передаче, и аудит.

# 84. Что такое роли в MongoDB?

Роли в MongoDB определяют набор привилегий, которые могут быть предоставлены пользователям для выполнения определенных действий над ресурсами базы данных.

# 85. Что такое привилегии в MongoDB?

Привилегии в MongoDB — это разрешения на выполнение конкретных операций (например, чтение, запись, создание индекса) над определенными ресурсами (например, коллекцией, базой данных).

# 86. Что такое аудит в MongoDB?

Аудит в MongoDB позволяет отслеживать и записывать административные действия и операции доступа к данным для анализа безопасности и соответствия требованиям.

# 87. Что такое шифрование в MongoDB?

Шифрование в MongoDB включает шифрование данных "в состоянии покоя" (на диске) и шифрование "в пути" (при передаче по сети с использованием TLS/SSL).

# 88. Что такое SSL/TLS в MongoDB?

SSL/TLS в MongoDB используется для шифрования сетевых соединений между клиентами и серверами, а также между членами реплика-сета и шардированного кластера.

# 89. Что такое LDAP в MongoDB?

Интеграция с LDAP в MongoDB позволяет использовать существующий LDAP-сервер для аутентификации пользователей и управления их доступом к базам данных.

# 90. Что такое Kerberos в MongoDB?

Интеграция с Kerberos в MongoDB позволяет использовать эту сетевую службу аутентификации для безопасного подтверждения личности пользователей и серверов.

# 91. Что такое x.509 в MongoDB?

Аутентификация с использованием сертификатов x.509 в MongoDB позволяет клиентам и серверам подтверждать свою личность с помощью цифровых сертификатов.

# 92. Что такое SCRAM в MongoDB?

SCRAM (Salted Challenge Response Authentication Mechanism) — это основной механизм аутентификации по умолчанию в MongoDB, обеспечивающий безопасное хранение и проверку паролей.

# 93. Что такое OAuth в MongoDB?

MongoDB напрямую не реализует OAuth как механизм аутентификации, но приложения, использующие MongoDB, могут интегрироваться с OAuth провайдерами для управления доступом пользователей.

# 94. Что такое JWT в MongoDB?

JWT (JSON Web Tokens) могут использоваться приложениями для аутентификации и авторизации запросов к MongoDB, где приложение проверяет токен и предоставляет доступ.

# 95. Что такое API в MongoDB?

API в MongoDB — это набор команд и протоколов, которые драйверы и инструменты используют для взаимодействия с базой данных, выполнения запросов и управления данными.

# 96. Что такое драйверы в MongoDB?

Драйверы MongoDB — это клиентские библиотеки для различных языков программирования, предоставляющие API для подключения к MongoDB и выполнения операций с базой данных.

# 97. Что такое MongoDB Atlas?

MongoDB Atlas — это полностью управляемый облачный сервис баз данных MongoDB, предоставляемый разработчиками MongoDB.

# 98. Что такое MongoDB Compass?

MongoDB Compass — это официальный графический интерфейс пользователя (GUI) для MongoDB, позволяющий визуализировать данные, выполнять запросы и оптимизировать производительность.

# 99. Что такое MongoDB Shell?

MongoDB Shell (`mongosh` или устаревший `mongo`) — это интерактивная JavaScript-консоль для взаимодействия с экземплярами MongoDB, администрирования и выполнения запросов.

# 100. Что такое MongoDB Charts?

MongoDB Charts — это инструмент для создания визуализаций и дашбордов на основе данных, хранящихся в MongoDB Atlas.

# 101. Что такое MongoDB Stitch?

MongoDB Stitch (теперь часть MongoDB Realm) предоставлял бэкенд-сервисы, такие как функции, триггеры и аутентификация, для упрощения разработки приложений.

# 102. Что такое MongoDB Realm?

MongoDB Realm — это платформа для разработки мобильных и веб-приложений, включающая синхронизацию данных с MongoDB Atlas, функции, триггеры и аутентификацию.

# 103. Что такое MongoDB Data API?

MongoDB Data API (часть Atlas App Services) позволяет безопасно обращаться к данным в MongoDB Atlas через стандартные HTTPS запросы без написания серверного кода.

# 104. Что такое MongoDB Query API?

MongoDB Query API — это богатый язык запросов, используемый для выборки, фильтрации, сортировки и проекции документов из коллекций MongoDB.

# 105. Что такое MongoDB Aggregation API?

MongoDB Aggregation API (Aggregation Pipeline) — это мощный фреймворк для выполнения многоэтапной обработки данных и вычислений над документами в коллекциях.