27. \*\*ORM\*\* (Object-Relational Mapping) - это техника программирования, которая позволяет связать объекты в программе с записями в базе данных, обеспечивая удобный способ доступа к данным. Он позволяет разработчикам использовать объектно-ориентированный подход к работе с данными, скрывая детали работы с базой данных, такие как SQL-запросы. ORM преобразует данные между объектами в программе и записями в базе данных, облегчая взаимодействие с данными.

28. \*\*Модель\*\* - это абстрактное представление данных в приложении. В контексте баз данных модель описывает структуру данных, включая сущности (таблицы), их атрибуты (поля) и отношения между сущностями. Модель определяет, как данные будут храниться и организованы в базе данных, и как они будут доступны для использования в приложении.

29. \*\*Пул соединений\*\* - это механизм, используемый базами данных для управления соединениями с клиентами (приложениями). Он представляет собой набор предварительно установленных соединений с базой данных, которые могут быть многократно использованы клиентами для выполнения запросов. Пул соединений позволяет эффективно управлять ресурсами базы данных, минимизируя накладные расходы на установку и разрыв соединения для каждого запроса.

30. \*\*Sequelize\*\* и \*\*Prisma\*\* - это два популярных ORM-фреймворка для работы с базами данных в приложениях на языке JavaScript. Sequelize поддерживает различные базы данных SQL, в то время как Prisma предоставляет современный подход к работе с данными и поддерживает несколько баз данных, включая SQL и NoSQL.

31. \*\*Scope\*\* в Sequelize - это специальный механизм, который позволяет определять наборы условий для запросов, которые могут быть повторно использованы в разных частях приложения.

32. \*\*Хуки\*\* (Hooks) в Sequelize - это функции, которые позволяют встраивать пользовательский код перед или после выполнения различных операций с моделью базы данных. Они бывают перед созданием, обновлением, удалением или извлечением данных.

33. Метод `Model.hasMany` в Sequelize используется для определения отношения "один ко многим" между моделями. Он позволяет указать, что одна модель может иметь несколько связанных записей в другой модели.

34. \*\*Raw query\*\* в Sequelize - это возможность выполнения SQL-запросов напрямую к базе данных без использования методов ORM. Он предоставляет более гибкий способ работы с базой данных, но требует более аккуратного обращения с данными и не обеспечивает некоторые преимущества ORM, такие как проверка типов и безопасность.

35. \*\*Prisma\*\* отличается от других фреймворков ORM своим современным подходом к работе с данными, автоматической генерацией типобезопасного клиентского кода и поддержкой нескольких баз данных, включая SQL и NoSQL.

36. Да, Prisma можно использовать как с базами данных SQL (например, PostgreSQL, MySQL) так и с NoSQL базами данных (например, MongoDB).

37. Prisma совместим с языками программирования JavaScript и TypeScript.

38. Методы `every`, `none` и `some` в Prisma используются для фильтрации результатов запросов с использованием логических операторов "и", "не" и "или" соответственно.

39. Метод `upsert` в Prisma используется для выполнения операции "обновить или вставить" - он обновляет запись в базе данных, если она существует, или вставляет новую запись, если она отсутствует.

40. Для включения регистронезависимого поиска в Prisma можно использовать функцию `toLowerCase()` или `toUpperCase()` внутри запроса для приведения строк к нижнему или верхнему регистру, соответственно.

41. Методы `$queryRaw` и `$executeRaw` в Prisma используются для выполнения произвольных SQL-запросов к базе данных. `$queryRaw` используется для выполнения запросов, которые возвращают результаты, в то время как `$executeRaw` используется для выполнения запросов, которые не возвращают результатов, таких как обновление или удаление данных.