

Тест 3 - Основы работы с БД - задачи даны в РНР.

- | | |
|-------|-------|
| 1. а | 11. а |
| 2. б | 12. д |
| 3. д | 13. б |
| 4. с | 14. а |
| 5. а | 15. а |
| 6. а | 16. а |
| 7. а | 17. а |
| 8. а | 18. а |
| 9. а | 19. б |
| 10. а | 20. б |

21. Prepared statements: механизм при котором SQL-запрос сначала отправляется на сервер без данных, а потом данные присоединяются отдельно. Тогда данные понимает, что это просто данные, а не часть команды, что ускорит выполнение запроса.

ного кода невозможны.

22. Транзакции: используемые методы `beginTransaction()`, `commit()` и `rollback()`. Если в процессе из-за переноса запросов второй этап, `rollback()` отменит изменение первого. Это спасает, например, при переводе денег.

23. Массовое добавление: лучше всего подготовить запрос один раз и выполнить его в цикле с разными данными.

24. Для безопасности пароли лучше хранить в файлах `.env` или конфигурационных файлах за пределами папки `public` сайта.

25. `mysqli` работает только с MySQL. PDO поддерживает 12 разных баз данных. PDO предпочтительнее для гибкости и профессиональной разра-

Сумми.

26. Условные fetch() в while вместе fetch till(). Так упрощают суммирование по одной строке, не переписывая операторов на языке сервера.

27. JOIN пример:

```
SELECT users.name, orders.amount  
FROM users JOIN orders ON users.  
.id = orders.user.id;
```

28. Memory \$start -> fetch() и
\$start -> fetchColumn()

29. Оптимизация запросов: создание индексов, отказ от SELECT *, копирование результатов.

30. Индекс - это как алфавитный указатель в конце книги. Без него база ищет строку перебором всей таблицы, что неэффективно и требует много времени.

сравниваю.

31. c

36. b

41. d

32. b

37. a

42. a

33. a

38. a

43. a

34. d

39. a

44. a

35. c

40. a

45. c

46. Сним мышцы: функциональный снимок (мышцы connect) привнесен тем, кто перешел с PHP 4. Объектный (перо. мышцы) - более современный, чистый и позволяет легче масштабировать код.

48. Многократная вставка: подготовим INSERT один раз, затем в цикле несколько перемещаемся, привязывая через bind-параметры и вызывая execute()

49. Транзакции в мышцах реализуются через \$mysqli → begin-transaction(), \$mysqli → commit() и

\$mySql → callback(). Позволяет
избегать "мусора" в базе при
частичном выполнении кода.

50. Кэширование — это сохранение
результатов текущих запросов
в памяти. Это позволяет не
мучить базу данными описания
тех же вопросах 1000 раз в се-
кунду.