**Лабораторная работа №2: Тестирование PostgreSQL**

**1. Цель работы**

Цель работы – изучение классификации видов тестирования и применение ее для анализа процесса тестирования системы управления базами данных PostgreSQL.

**2. Описание программного проекта**

**Программный проект:** Система управления базами данных PostgreSQL.

**Краткое описание:** PostgreSQL – это объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом. Она отличается высокой надежностью, стабильностью и богатым функционалом, включая поддержку транзакций, ACID-свойств, SQL-стандарт, а также расширяемость.

**3. Классификация видов тестирования для PostgreSQL**

**3.1. Уровни тестирования:**

* **Компонентное (модульное) тестирование:**
  + Проверка отдельных модулей PostgreSQL, таких как:
    - Модуль парсера SQL-запросов (parser)
    - Модуль планировщика запросов (planner)
    - Модуль выполнения запросов (executor)
    - Модуль управления транзакциями (transaction manager)
  + Проверка правильности обработки данных, выполнения операций (CRUD), корректной работы алгоритмов и функций.
  + Примеры дефектов: ошибки синтаксического анализа SQL-запросов, некорректное планирование запросов, неверные результаты выполнения операций.
* **Интеграционное тестирование:**
  + Проверка взаимодействия между различными компонентами PostgreSQL, такими как:
    - Взаимодействие модуля парсера с модулем планировщика
    - Взаимодействие модуля планировщика с модулем выполнения
    - Взаимодействие модуля управления транзакциями с другими модулями
    - Взаимодействие с внешними системами, например, с драйверами JDBC, ODBC.
  + Проверка корректной передачи данных между модулями, согласованности работы компонентов, правильной обработки транзакций.
  + Примеры дефектов: ошибки передачи данных между модулями, несогласованность состояний данных, нарушения ACID-свойств.
* **Системное тестирование:**
  + Проверка работоспособности PostgreSQL как цельной системы, включая:
    - Проверка соответствия требованиям к производительности, надежности, безопасности.
    - Проверка работы в различных конфигурациях (разные операционные системы, аппаратные платформы, объемы данных).
    - Проверка функциональности основных возможностей, таких как создание, изменение, удаление баз данных, таблиц, запросы, триггеры, представления.
  + Примеры дефектов: проблемы с производительностью, ошибки в работе с большим объемом данных, уязвимости в безопасности, некорректная работа с внешними системами.

**3.2. Типы тестирования:**

* **Функциональное тестирование:**
  + Проверка выполнения всех основных функций PostgreSQL, включая:
    - Создание и удаление баз данных, таблиц, представлений
    - Ввод, редактирование и удаление данных
    - Выполнение запросов (SELECT, INSERT, UPDATE, DELETE)
    - Работа с транзакциями, блокировками, триггерами
  + Проверка корректности результатов выполнения операций, соответствия результатам ожидаемым результатам, соответствие SQL-стандарту.
  + Примеры дефектов: ошибки при выполнении SQL-запросов, некорректное выполнение операций, нарушение ограничений целостности данных.
* **Тестирование безопасности:**
  + Проверка устойчивости PostgreSQL к атакам и уязвимостям.
  + Проверка механизмов аутентификации, авторизации, шифрования данных.
  + Проверка защищенности от SQL-инъекций, атак на отказ в обслуживании, и других угроз.
  + Примеры дефектов: уязвимости в механизмах аутентификации, возможность SQL-инъекций, слабая защита от атак на отказ в обслуживании.
* **Тестирование производительности:**
  + Проверка производительности PostgreSQL в различных условиях:
    - Проверка времени обработки запросов, пропускной способности системы, потребления ресурсов (CPU, память).
    - Проверка работы PostgreSQL при различной нагрузке, объеме данных, количестве пользователей.
  + Примеры дефектов: медленная обработка запросов, перегрузка ресурсов, сбои при высоких нагрузках.
* **Тестирование удобства использования:**
  + Проверка удобства работы с PostgreSQL для администраторов и разработчиков.
  + Проверка интерфейса командной строки, графических инструментов администрирования, документации.
  + Примеры дефектов: неудобный интерфейс, сложная навигация, неполная или некорректная документация.

**3.3. Методы тестирования:**

* **Статические методы:**
  + **Рецензирование кода:**
    - Проверка кода PostgreSQL на соответствие стандартам, выявление потенциальных ошибок, оценка качества кода.
  + **Статический анализ:**
    - Применение инструментов статического анализа для выявления потенциальных ошибок, уязвимостей в коде PostgreSQL.
* **Динамические методы:**
  + **Тестирование черного ящика:**
    - Проверка функций PostgreSQL без знания внутреннего устройства.
    - Создание тестовых сценариев на основе функциональных спецификаций.
  + **Тестирование белого ящика:**
    - Проверка функций PostgreSQL с учетом знания внутреннего устройства.
    - Создание тестовых сценариев с учетом внутренних механизмов работы системы.
  + **Тестирование серого ящика:**
    - Комбинирование методов черного и белого ящика.
    - Проверка функций с частичным знанием внутреннего устройства.

**3.4. Виды тестирования:**

* **Тестирование функциональности:**
  + Проверка корректной работы основных функций PostgreSQL: создание, изменение, удаление баз данных, таблиц, запросов, триггеров, представлений.
* **Тестирование установки:**
  + Проверка корректной работы инсталляционного процесса PostgreSQL, включая проверку установки на разных операционных системах.
* **Тестирование нагрузочное:**
  + Проверка поведения PostgreSQL при высоких нагрузках, оценка производительности, проверка устойчивости к отказам.
* **Тестирование стресс:**
  + Проверка поведения PostgreSQL в условиях экстремальных нагрузок, оценка устойчивости к сбоям, проверка возможности восстановления после отказов.
* **Тестирование удобства использования (юзабилити):**
  + Проверка удобства работы с PostgreSQL для администраторов и разработчиков, оценка интерфейса, документации.
* **Тестирование совместимости:**
  + Проверка работы PostgreSQL с различными операционными системами, аппаратными платформами, версиями драйверов.
* **Тестирование безопасности:**
  + Проверка защищенности PostgreSQL от атак, оценка механизмов аутентификации, авторизации, шифрования данных.

**3.5. Субъект тестирования:**

* **Внутреннее тестирование:**
  + Тестирование, проводимое разработчиками PostgreSQL.
* **Внешнее тестирование:**
  + Тестирование, проводимое независимыми компаниями или пользователями.

**3.6. По степени автоматизации:**

* **Ручное тестирование:**
  + Выполнение тестовых сценариев вручную.
* **Автоматизированное тестирование:**
  + Использование автоматических инструментов для выполнения тестовых сценариев.
* **Полуавтоматизированное тестирование:**
  + Комбинирование ручного и автоматизированного тестирования.

**3.7. По признаку позитивности:**

* **Позитивное тестирование:**
  + Проверка системы на корректность работы в штатном режиме.
* **Негативное тестирование:**
  + Проверка поведения системы при некорректных входных данных, ошибках, сбоях.

**3.8. По времени проведения:**

* **Альфа-тестирование:**
  + Тестирование, проводимое разработчиками PostgreSQL на ранних стадиях разработки.
* **Бета-тестирование:**
  + Тестирование, проводимое пользователями PostgreSQL перед выпуском финальной версии.
* **Приемочное тестирование:**
  + Тестирование, проводимое заказчиком перед принятием системы в эксплуатацию.
* **Регрессионное тестирование:**
  + Тестирование, проводимое после внесения изменений в PostgreSQL для проверки работоспособности предыдущих функций.

**4. Примеры дефектов**

**Компонентное тестирование:**

* Ошибка синтаксического анализа SQL-запросов.
* Некорректное планирование запросов.
* Неверные результаты выполнения операций CRUD.

**Интеграционное тестирование:**

* Ошибка передачи данных между модулем парсера и модулем планировщика.
* Несогласованность состояний данных между различными модулями.
* Нарушение ACID-свойств при работе с транзакциями.

**Системное тестирование:**

* Проблемы с производительностью PostgreSQL при высоких нагрузках.
* Сбои в работе системы при большом объеме данных.
* Уязвимости в безопасности.

**Функциональное тестирование:**

* Ошибки при выполнении SQL-запросов.
* Некорректное выполнение операций CRUD.
* Нарушение ограничений целостности данных.

**Тестирование безопасности:**

* Уязвимости в механизмах аутентификации.
* Возможность SQL-инъекций.
* Слабая защита от атак на отказ в обслуживании.

**Тестирование производительности:**

* Медленная обработка запросов.
* Перегрузка ресурсов.
* Сбои при высоких нагрузках.

**Тестирование удобства использования:**

* Неудобный интерфейс.
* Сложная навигация.
* Неполная или некорректная документация.

**5. Выводы по работе**

В ходе лабораторной работы изучена классификация видов тестирования и ее применение для анализа процесса тестирования системы управления базами данных PostgreSQL.

* Определены виды, типы, методы, уровни тестирования, применяемые для проверки PostgreSQL.
* Выявлены типичные дефекты, возникающие при тестировании PostgreSQL на различных уровнях.

Полученные знания позволят более эффективно планировать, проводить и анализировать процесс тестирования PostgreSQL, повышая качество разрабатываемого программного обеспечения.

**6. Список использованных источников**

* [Официальный сайт PostgreSQL](https://www.postgresql.org/)
* [Документация PostgreSQL](https://www.postgresql.org/docs/)
* [Книга "Тестирование программного обеспечения"](https://www.amazon.com/Testing-Software-Third-Edition-Effective/dp/0321113428)
* [Книга "Рефакторинг"](https://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0201485672)