



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования

«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФГБОУ ВО «ИГУ»

Факультет бизнес-коммуникаций и информатики

Кафедра естественнонаучных дисциплин

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине Б1.В.10 Базы данных (продвинутый уровень)

направление подготовки 09.03.03 Прикладная информатика

направленность (профиль) Прикладная информатика (разработка программного обеспечения)

Одобрено
УМК факультета бизнес-коммуникаций
и информатики

Разработан в соответствии с ФГОС ВО

с учетом требований проф. стандарта

Председатель УМК

М.Г. Синчурина

ФИО, должность, ученая степень, звание

подпись, печать

Разработчики:



(подпись)

доцент

(занимаемая должность)

А.Г. Балахчи

(инициалы, фамилия)

преподаватель

(занимаемая должность)

А.С. Воловиков

(инициалы, фамилия)

Цель фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Б1.В.10 Базы данных (продвинутый уровень)». Перечень видов оценочных средств соответствует рабочей программе дисциплины.

Фонд оценочных средств включает контрольные материалы для проведения текущего контроля (в следующих формах: тест, практическое задание, проект) и промежуточной аттестации в форме вопросов и заданий к зачету с оценкой.

Структура и содержание заданий – задания разработаны в соответствии с рабочей программой дисциплины «Б1.В.10 Базы данных (продвинутый уровень)».

1. Паспорт фонда оценочных средств

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-2 ППК-Р2 Способен проверять работоспособность и проводить рефакторинг кода программного обеспечения	ПК-2 ППК-Р2.1 Разрабатывает тестовые наборы данных для проверки работоспособности компьютерного программного обеспечения	• ПК-2 ППК-Р2.1 З-1. Знает методы создания и документирования контрольных примеров и тестовых наборов данных • ПК-2 ППК-Р2.1 З-2. Знает требования к структуре и форматам хранения тестовых наборов данных • ПК-2 ППК-Р2.1 З-3. Знает правила, алгоритмы и технологии создания тестовых наборов данных • ПК-2 ППК-Р2.1 У-1. Умеет разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения • ПК-2 ППК-Р2.1 У-2. Умеет готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения
	ПК-2 ППК-Р2.2 Проверяет работоспособность компьютерного программного обеспечения	• ПК-2 ППК-Р2.2 З-1. Знает методы и средства проверки работоспособности компьютерного программного обеспечения • ПК-2 ППК-Р2.2 З-2. Знает государственные стандарты испытания автоматизированных систем • ПК-2 ППК-Р2.2 З-3. Знает руководящие документы по стандартизации требований к документам автоматизированных систем • ПК-2 ППК-Р2.2 У-1. Умеет применять методы и средства проверки работоспособности компьютерного программного обеспечения • ПК-2 ППК-Р2.2 У-2. Умеет интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения • ПК-2 ППК-Р2.2 У-3. Умеет анализировать значения полученных характеристик компьютерного программного обеспечения
	ПК-2 ППК-Р2.3 Исправляет дефекты программного кода, зафиксированных в базе данных дефектов	• ПК-2 ППК-Р2.3 З-1. Знает типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления • ПК-2 ППК-Р2.3 З-2. Знает методы и приемы отладки программного кода • ПК-2 ППК-Р2.3 У-1. Умеет воспроизводить дефекты программного кода, зафиксированные в базе данных дефектов • ПК-2 ППК-Р2.3 У-2. Умеет выяснять причины возникновения дефектов программного кода • ПК-2 ППК-Р2.3 У-3. Умеет вносить изменений в программный код для устранения выявленных дефектов
	ПК-2 ППК-Р2.4 Выполняет рефакторинг и инспекцию программного кода (код-ревью)	• ПК-2 ППК-Р2.4 З-1. Знает методы и средства рефакторинга и инспекции программного кода • ПК-2 ППК-Р2.4 З-2. Знает нормативно-технические документы (стандарты и регламенты), регламентирующие требования к программному коду, порядок отражения изменений в системе управления версиями, порядок отражения результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний • ПК-2 ППК-Р2.4 У-1. Умеет анализировать программный код на соответствие требованиям по читаемости и производительности • ПК-2 ППК-Р2.4 У-2. Умеет проводить инспекцию программного кода для поиска не обнаруженных на ранних стадиях разработки компьютерного программного обеспечения ошибок и критических мест • ПК-2 ППК-Р2.4 У-3. Умеет применять методы и средства рефакторинга и инспекции программного кода • ПК-2 ППК-Р2.4 У-4. Умеет публиковать результаты рефакторинга и инспекции в коллективной базе знаний

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-3 ППК-Р3 Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта	ПК-3 ППК-Р3.1 Разрабатывает процедуры интеграции программных модулей	• ПК-3 ППК-Р3.1 З-1. Знает методы и средства сборки модулей и компонентов компьютерного программного обеспечения • ПК-3 ППК-Р3.1 З-2. Знает интерфейсы взаимодействия с внешней средой • ПК-3 ППК-Р3.1 З-3. Знает интерфейсы взаимодействия внутренних модулей системы • ПК-3 ППК-Р3.1 У-1. Умеет писать программный код процедур интеграции программных модулей • ПК-3 ППК-Р3.1 У-2. Умеет использовать выбранную среду программирования для разработки процедур интеграции программных модулей • ПК-3 ППК-Р3.1 У-3. Умеет применять методы и средства сборки модулей и компонентов компьютерного программного обеспечения, разработки процедур для развертывания компьютерного программного обеспечения, миграции и преобразования данных, создания программных интерфейсов
	ПК-3 ППК-Р3.2 Выполняет интеграцию программных модулей и компонентов и проверку работоспособности выпусков программного продукта	• ПК-3 ППК-Р3.2 З-1. Знает методы и средства сборки и интеграции программных модулей и компонентов • ПК-3 ППК-Р3.2 З-2. Знает методы и средства проверки работоспособности выпусков программных продуктов • ПК-3 ППК-Р3.2 У-1. Умеет выполнять процедуры сборки программных модулей и компонентов в программный продукт • ПК-3 ППК-Р3.2 У-2. Умеет производить настройки параметров программного продукта и осуществлять запуск процедур сборки • ПК-3 ППК-Р3.2 У-3. Умеет проводить проверку работоспособности программного продукта
ПК-4 ППК-Р4 Способен проектировать и интегрировать базы данных в программные модули и компоненты	ПК-4 ППК-Р4.1 Пишет программный код с использованием языков определения и манипулирования данными в базах данных	• ПК-4 ППК-Р4.1 З-1. Знает архитектуры современных систем управления баз данных, включая SQL и noSQL • ПК-4 ППК-Р4.1 З-2. Знает синтаксис языка работы с выбранной базой данных, особенности программирования на этом языке • ПК-4 ППК-Р4.1 З-3. Знает современные среды программирования для работы с базами данных • ПК-4 ППК-Р4.1 У-1. Умеет применять выбранные языки работы с базами данных • ПК-4 ППК-Р4.1 У-2. Умеет использовать выбранную среду программирования для работы с данными в базе • ПК-4 ППК-Р4.1 У-3. Умеет использовать методы и средства выбранного языка программирования для работы с базами данных
	ПК-4 ППК-Р4.2 Проектирует базы данных для программных модулей и компонентов	• ПК-4 ППК-Р4.2 З-1. Знает современные подходы к проектированию реляционных и нереляционных баз данных • ПК-4 ППК-Р4.2 З-2. Знает нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению моделей баз данных выбранной архитектуры • ПК-4 ППК-Р4.2 У-1. Умеет выбирать тип базы данных в зависимости от решаемой задачи • ПК-4 ППК-Р4.2 У-2. Умеет проектировать и актуализировать структуру базы данных для программных моделей и компонентов • ПК-4 ППК-Р4.2 У-3. Умеет применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению модели баз данных • ПК-4 ППК-Р4.2 У-4. Умеет применять инструментальный для создания и актуализации моделей баз данных
	ПК-4 ППК-Р4.3 Оптимизирует производительность работы с базами данных	• ПК-4.3 ППК-Р4.3 З-1. Знает внутреннее устройство СУБД выбранной архитектуры • ПК-4.3 ППК-Р4.3 З-2. Знает методы и средства мониторинга и оптимизации производительности СУБД выбранной архитектуры • ПК-4.3 ППК-Р4.3 У-1. Умеет применять методы и средства мониторинга производительности запросов к базе данных • ПК-4.3 ППК-Р4.3 У-2. Умеет вырабатывать варианты оптимизации производительности запросов в базе данных • ПК-4.3 ППК-Р4.3 У-3. Умеет проводить оценку и обоснование рекомендуемых решений по оптимизации производительности запросов в базе данных

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-5 ППК-Р5 Способен разрабатывать требования и проектировать программное обеспечение	ПК-5 ППК-Р5.1 Анализирует возможности реализации требований к компьютерному программному обеспечению	• ПК-5 ППК-Р5.1 З-1. Знает возможности существующей программно-технической архитектуры • ПК-5 ППК-Р5.1 З-2. Знает возможности современных и перспективных средств разработки программных продуктов, технических средств • ПК-5 ППК-Р5.1 З-3. Знает методологии разработки компьютерного программного обеспечения и технологии программирования • ПК-5 ППК-Р5.1 У-1. Умеет проводить сбор и систематизацию требований к компьютерному программному обеспечению • ПК-5 ППК-Р5.1 У-2. Умеет выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению • ПК-5 ППК-Р5.1 У-3. Умеет вырабатывать варианты реализации требований к компьютерному программному обеспечению
	ПК-5 ППК-Р5.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	• ПК-5 ППК-Р5.2 З-1. Знает методы и средства проектирования компьютерного программного обеспечения • ПК-5 ППК-Р5.2 З-2. Знает методы и средства проектирования программных интерфейсов • ПК-5 ППК-Р5.2 У-1. Умеет выбирать средства реализации требований к компьютерному программному обеспечению • ПК-5 ППК-Р5.2 У-2. Умеет вырабатывать варианты реализации компьютерного программного обеспечения • ПК-5 ППК-Р5.2 У-3. Умеет проводить оценку и обоснование рекомендуемых решений
	ПК-5 ППК-Р5.3 Проектирует компьютерное программное обеспечение	• ПК-5 ППК-Р5.3 З-1. Знает принципы построения и виды архитектуры компьютерного программного обеспечения • ПК-5 ППК-Р5.3 З-2. Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения • ПК-5 ППК-Р5.3 З-3. Знает нормативно-технические документы (стандарты), определяющие требования к технической документации на компьютерное программное обеспечение • ПК-5 ППК-Р5.3 З-4. Знает методы и средства проектирования компьютерного программного обеспечения • ПК-5 ППК-Р5.3 У-1. Умеет разрабатывать и изменять архитектуру компьютерного программного обеспечения и согласовывать ее с системным аналитиком и архитектором программного обеспечения • ПК-5 ППК-Р5.3 У-2. Умеет проектировать структуры данных • ПК-5 ППК-Р5.3 У-3. Умеет проектировать программные интерфейсы
	ПК-5 ППК-Р5.4 Выполняет логическое проектирование системы	• ПК-5 ППК-Р5.4 З-1. Знает устройство и функционирование ИТ-систем/продуктов • ПК-5 ППК-Р5.4 З-2. Знает методы моделирования и описания устройства и функционирования ИТ-систем/продуктов, их частей, обеспечения и окружения • ПК-5 ППК-Р5.4 З-3. Знает базовые технологии взаимодействия и интеграции систем и компонентов • ПК-5 ППК-Р5.4 З-4. Знает методы функциональной декомпозиции ИТ-систем • ПК-5 ППК-Р5.4 У-1. Умеет декомпозировать ИТ-системы и ИТ-продукты на подсистемы и элементы поставки • ПК-5 ППК-Р5.4 У-2. Умеет описывать интерфейсы пользователя на логическом уровне • ПК-5 ППК-Р5.4 У-3. Умеет описывать интеграции со смежными системами на логическом уровне

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-6 ИП-2 Способен оптимизировать производительность программного обеспечения	ПК-6 ИП-2.1 Выполняет мониторинг производительности программного обеспечения	• ПК-6 ИП-2.1 З-1. Знает методы и средства мониторинга производительности компьютерного программного обеспечения • ПК-6 ИП-2.1 З-2. Знает метрики производительности программного обеспечения • ПК-6 ИП-2.1 З-3. Знает современные инструменты мониторинга производительности программного обеспечения • ПК-6 ИП-2.1 У-1. Умеет применять методы и средства мониторинга производительности компьютерного программного обеспечения • ПК-6 ИП-2.1 У-2. Умеет интерпретировать диагностические данные мониторинга производительности компьютерного программного обеспечения • ПК-6 ИП-2.1 У-3. Умеет определять "узкие места" программного обеспечения
	ПК-6 ИП-2.2 Выполняет оптимизацию программного кода	• ПК-6 ИП-2.2 З-1. Знает методы и средства оптимизации производительности компьютерного программного обеспечения • ПК-6 ИП-2.2 З-2. Знает современные инструменты оптимизации производительности программного обеспечения • ПК-6 ИП-2.2 У-1. Умеет оптимизировать программный код с использованием специализированных программных средств • ПК-6 ИП-2.2 У-2. Умеет вырабатывать варианты оптимизации производительности компьютерного программного обеспечения • ПК-6 ИП-2.2 У-3. Умеет проводить оценку и обоснование рекомендуемых решений по оптимизации производительности
ПК-7 ППК-Р6 Способен участвовать в промышленной разработке программного обеспечения	ПК-7 ППК-Р6.1 Работает в соответствии с промышленными методологиями разработки	• ПК-7 ППК-Р6.1 З-1. Знает принципы Agile и их применение в промышленных проектах • ПК-7 ППК-Р6.1 З-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership) • ПК-7 ППК-Р6.1 У-1. Умеет оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов • ПК-7 ППК-Р6.1 У-2. Умеет работать в команде с использованием инструментов управления проектами
	ПК-7 ППК-Р6.2 Использует инструменты промышленной разработки	• ПК-7 ППК-Р6.2 З-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD) • ПК-7 ППК-Р6.2 З-2. Знает системы мониторинга и логирования в продуктивной среде • ПК-7 ППК-Р6.2 У-1. Умеет настраивать потоки работ CI/CD • ПК-7 ППК-Р6.2 У-2. Умеет работать с контейнеризацией и оркестрацией • ПК-7 ППК-Р6.2 У-3. Умеет настраивать мониторинг в продуктивной среде
	ПК-7 ППК-Р6.3 Разрабатывает масштабируемый и поддерживаемый код	• ПК-7 ППК-Р6.3 З-1. Знает принципы чистого кода, SOLID, DRY, KISS и др • ПК-7 ППК-Р6.3 З-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения • ПК-7 ППК-Р6.3 З-3. Знает паттерны проектирования и антипаттерны • ПК-7 ППК-Р6.3 У-1. Умеет разрабатывать модульный и тестируемый программный код • ПК-7 ППК-Р6.3 У-2. Умеет выполнять модульное, интеграционное и нагрузочное тестирование • ПК-7 ППК-Р6.3 У-3. Умеет проводить рефакторинг для повышения качества кода • ПК-7 ППК-Р6.3 У-4. Умеет применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах
	ПК-7 ППК-Р6.4 Участствует в развертывании и поддержке программного обеспечения	• ПК-7 ППК-Р6.4 З-1. Знает стратегии развертывания промышленного программного обеспечения • ПК-7 ППК-Р6.4 З-2. Знает основы работы с облачными платформами • ПК-7 ППК-Р6.4 У-1. Умеет развертывать приложения в облаке • ПК-7 ППК-Р6.4 У-2. Умеет обнаруживать и устранять инциденты с работой продуктивного программного обеспечения

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-8 ППК-Р7 Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода	ПК-8 ППК-Р7.1 Применяет ИИ-инструменты для генерации программного кода	• "ПК-8 ППК-Р7.1 З-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода • ПК-8 ППК-Р7.1 З-2. Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование) • ПК-8 ППК-Р7.1 У-1. Умеет формулировать корректные текстовые запросы (промты) для генерации кода • ПК-8 ППК-Р7.1 У-2. Умеет интегрировать ИИ-инструменты в среду разработки."
	ПК-8 ППК-Р7.2 Использует ИИ для анализа и отладки кода	• ПК-8 ППК-Р7.2 З-1. Знает методы ИИ-анализа кода • ПК-8 ППК-Р7.2 З-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ • ПК-8 ППК-Р7.2 У-1. Умеет настраивать ИИ-инструменты для поиска уязвимостей • ПК-8 ППК-Р7.2 У-2. Умеет интерпретировать рекомендации ИИ по исправлению кода
	ПК-8 ППК-Р7.3 Оптимизирует код с помощью ИИ	• ПК-8 ППК-Р7.3 З-1. Знает методы ИИ-оптимизации • ПК-8 ППК-Р7.3 З-2. Знает критерии качества кода, применяемые ИИ-системами • ПК-8 ППК-Р7.3 У-1. Умеет использовать ИИ для рефакторинга • ПК-8 ППК-Р7.3 У-2. Умеет проверять корректность оптимизаций, предложенных ИИ
	ПК-8 ППК-Р7.4 Оценивает этические и профессиональные аспекты применения ИИ в разработке	• ПК-8 ППК-Р7.4 З-1. Знает этические нормы использования ИИ (конфиденциальность, плагиат кода и т.п.) • ПК-8 ППК-Р7.4 З-2. Знает лицензионные ограничения сгенерированного кода • ПК-8 ППК-Р7.4 У-1. Умеет проверять код на соответствие стандартам после ИИ-генерации • ПК-8 ППК-Р7.4 У-2. Умеет документировать использование ИИ в разработке

2. Показатели и критерии оценивания компетенций, шкалы оценивания

2.1. Показатели и критерии оценивания компетенций

№ п/п	Раздел, тема	Код индикатора компетенции	Наименование ОС	
			ТК	ПА
1	Подготовка	ПК-4 ППК-Р4.1, ПК-3 ППК-Р3.1, ПК-3 ППК-Р3.2, ПК-4 ППК-Р4.2, ПК-5 ППК-Р5.1, ПК-5 ППК-Р5.2, ПК-5 ППК-Р5.3, ПК-5 ППК-Р5.4	Тест, Пз, Проект	Тест
2	Создание скриптов	ПК-4 ППК-Р4.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-2 ППК-Р2.4, ПК-4 ППК-Р4.3, ПК-8 ППК-Р7.1, ПК-8 ППК-Р7.2, ПК-8 ППК-Р7.3, ПК-8 ППК-Р7.4	Тест, Пз	Тест
3	Альтернативный расчет	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4	Тест, Пз	Тест
4	Наполнение данными	ПК-4 ППК-Р4.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-2 ППК-Р2.4, ПК-3 ППК-Р3.1, ПК-3 ППК-Р3.2, ПК-4 ППК-Р4.2, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-8 ППК-Р7.1, ПК-8 ППК-Р7.2, ПК-8 ППК-Р7.3	Тест, Пз	Тест

№ п/п	Раздел, тема	Код индикатора компетенции	Наименование ОС	
			ТК	ПА
5	Использование массивов	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-8 ППК-Р7.1, ПК-8 ППК-Р7.2, ПК-8 ППК-Р7.3	Тест, Пз	Тест
6	Использование процедур	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4	Тест, Пз	Тест
7	Разработка универсальных структур	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4, ПК-3 ППК-Р3.2, ПК-7 ППК-Р6.1, ПК-7 ППК-Р6.3	Тест, Пз	Тест
8	Расчеты	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4	Тест, Пз	Тест
9	Триггеры	ПК-4 ППК-Р4.1, ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4	Тест, Пз	Тест
10	Финал	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4, ПК-3 ППК-Р3.2, ПК-5 ППК-Р5.2, ПК-5 ППК-Р5.3, ПК-5 ППК-Р5.4, ПК-7 ППК-Р6.1, ПК-7 ППК-Р6.3, ПК-7 ППК-Р6.4, ПК-8 ППК-Р7.2, ПК-8 ППК-Р7.3, ПК-8 ППК-Р7.4	Тест, Пз	Тест

2.2. Критерии оценивания результатов обучения для текущего контроля успеваемости и промежуточной аттестации

Оценочное средство	Критерии оценивания	Шкала оценивания
Тест	Студентом даны правильные ответы на 91-100% заданий	Отлично
	Студентом даны правильные ответы на 81-90% заданий	Хорошо
	Студентом даны правильные ответы на 71-80% заданий	Удовлетворительно
	Студентом даны правильные ответы менее чем на 70% заданий	Неудовлетворительно

Оценочное средство	Критерии оценивания	Шкала оценивания
Практическое задание	Задание выполнено верно. Выбран оптимальный путь решения. Присутствует развернутое описание алгоритма решения	Отлично
	Задание выполнено верно. Допущены негрубые логические ошибки при описании алгоритма решения. Отсутствуют пояснения к решению задания	Хорошо
	Ход решения задания верный, но допущены ошибки приведшие к неправильному ответу	Удовлетворительно
	В работе получен неверный ответ, связанный с грубыми ошибками допущенными в ходе решения, либо решение отсутствует полностью	Неудовлетворительно

Оценочное средство	Критерии оценивания	Шкала оценивания
Проект	Проект студентом завершён в полном объеме. Для естественнонаучного проекта представлена работоспособная практическая часть, правильно выполнены и обоснованы необходимые расчёты. Реализация практической части проекта соответствует техническому заданию или заданию преподавателя. В проекте обозначена актуальность выбора темы с опорой на анализ предметной области. Студент способен сформулировать и обосновать практическую значимость своей работы. Подготовлена презентация результатов работы. Студент ориентируется во всех этапах разработки проекта, уверенно отвечает на вопросы аудитории. Способен аргументированно обосновать концепцию проекта и выбор инструментов для разработки проекта. Студент способен выделить достоинства и недостатки своей работы и предложить способы устранения недостатков	Отлично
	Проект студентом в целом завершён. Выполнены ключевые задачи. Для естественнонаучного проекта практическая часть в целом работоспособна, но есть мелкие неустранённые недостатки, необходимые расчеты в целом выполнены верно, но есть небольшие замечания. Реализация проекта в целом соответствует техническому заданию или заданию преподавателя. В проекте обозначена, но недостаточно обоснована актуальность темы. Практическая значимость работы просматривается, студент в целом может её сформулировать. Подготовлена презентация результатов работы. Студент ориентируется в этапах разработки проекта, но отвечает не на все вопросы аудитории. Способен обосновать выбор инструментов для реализации проекта. Студент способен выделить достоинства и недостатки своей работы, но не может предложить способы устранения последних	Хорошо
	Проект студентом завершён только в основных пунктах задания. Выполнены только ключевые задачи с недостатками. Для естественнонаучного проекта практическая часть работоспособна не вполне, есть существенные неустранённые недостатки, необходимые расчеты выполнены с ошибками. Реализация проекта частично соответствует техническому заданию или заданию преподавателя. В проекте обозначена, но не обоснована актуальность темы. Практическая значимость работы сформулирована слабо. Подготовлена презентация результатов работы. Студент слабо ориентируется в этапах разработки проекта, отвечает только на некоторые вопросы аудитории. Плохо обосновывает выбор инструментов для реализации проекта. Студент способен выделить достоинства и недостатки своей работы после серии наводящих вопросов, но не может предложить способы устранения недостатков	Удовлетворительно
	Проект студентом не завершён. Неполностью выполнены или не выполнены совсем ключевые задачи. Для естественнонаучного проекта практическая часть не работоспособна или не начата, есть существенные неустранённые недостатки, необходимые расчеты выполнены с грубыми ошибками. Реализация проекта не соответствует техническому заданию или заданию преподавателя. В проекте необозначена, и не обоснована актуальность темы. Практическая значимость работы не сформулирована. Плохо подготовлена презентация результатов работы. Студент почти не ориентируется в этапах разработки проекта, не отвечает на вопросы аудитории. Плохо обосновывается выбор инструментов для реализации проекта. Студент не способен выделить достоинства и недостатки своей работы даже после серии наводящих вопросов	Неудовлетворительно

2.3. Оценочные средства для текущего контроля (примеры)

2.3.1. Материалы для компьютерного тестирования обучающихся

Общие критерии оценивания

Процент правильных ответов	Оценка
91% – 100%	5 (отлично)

Процент правильных ответов	Оценка
81% – 90%	4 (хорошо)
71% – 80%	3 (удовлетворительно)
Менее 70%	2 (неудовлетворительно)

Соответствие вопросов теста индикаторам формируемых и оцениваемых компетенций

№ вопроса в тесте	Код индикатора компетенции
1	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
2	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
3	ПК-4 ППК-Р4.1
4	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
5	ПК-4 ППК-Р4.1
6	ПК-4 ППК-Р4.1
7	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
8	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2
9	ПК-4 ППК-Р4.1
10	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4

Ключ ответов

№ вопроса в тесте	Номер ответа (или ответ, или соответствие)
1	b
2	d
3	d
4	c
5	a
6	d
7	c
8	c
9	a
10	d

Перечень тестовых вопросов

№ 1. Задание с единичным выбором. Выберите один правильный ответ.

Что такое PostgreSQL?

- a. Фреймворк для веб-разработки
- b. Система управления реляционными базами данных
- c. Инструмент для анализа больших данных
- d. Язык программирования

№ 2. Задание с единичным выбором. Выберите один правильный ответ.

Что вернёт следующий SQL-запрос в PostgreSQL?

- a. Все записи из таблицы сотрудников (employees)
- b. Все записи из таблицы отделов (departments)
- c. Только общие поля обеих таблиц
- d. Произведение каждой строки из таблицы сотрудников на каждую строку из таблицы отделов

№ 3. Задание с единичным выбором. Выберите один правильный ответ.

Когда выполняется триггер AFTER INSERT в PostgreSQL?

- a. Перед вставкой новой записи в таблицу
- b. После неудачной попытки вставки новой записи
- c. Во время проверки ограничений целостности перед вставкой
- d. После успешной вставки новой записи в таблицу

№ 4. Задание с единичным выбором. Выберите один правильный ответ.

Как правильно объявляется локальная переменная внутри блока PL/pgSQL в PostgreSQL?

- a. VAR <переменная> := <тип>;
- b. <переменная> AS <тип>;
- c. DECLARE <переменная> <тип>;
- d. DECLARE <переменная> TYPE <тип>;

№ 5. Задание с единичным выбором. Выберите один правильный ответ.

Что обозначают аббревиатуры DML в контексте PostgreSQL?

- a. Data Manipulation Language (язык манипулирования данными)
- b. Database Management Layer (уровень управления базой данных)
- c. Dynamic Metadata Library (динамическая библиотека метаданных)
- d. Distributed Multi-level Locking (распределённая многоуровневая блокировка)

№ 6. Задание с единичным выбором. Выберите один правильный ответ.

Какая последовательность команд в PostgreSQL запускает транзакцию, записывает изменения и подтверждает их выполнение?

- a. DECLARE → SELECT → END
- b. SAVEPOINT → DELETE → RELEASE
- c. START TRANSACTION → INSERT → ROLLBACK
- d. BEGIN TRANSACTION → UPDATE → COMMIT

№ 7. Задание с единичным выбором. Выберите один правильный ответ.

Как называется специальная структура, используемая для передачи параметров внутрь процедуры в PostgreSQL?

- a. Cursor
- b. Argument
- c. Parameter
- d. Trigger

№ 8. Задание с единичным выбором. Выберите один правильный ответ.

Какая конструкция SQL-запроса позволяет создать столбец массива строковых значений в таблице PostgreSQL?

- a. column_name STRING_ARRAY
- b. column_name ARRAY(VARCHAR)
- c. column_name VARCHAR[]
- d. column_name []VARCHAR

№ 9. Задание с единичным выбором. Выберите один правильный ответ.

Что означает кнопка «Vacuum Analyze» в интерфейсе pgAdmin?

- a. Запускает процедуру оптимизации и сбора статистики для улучшения производительности запросов
- b. Выполняет резервное копирование всей базы данных
- c. Полностью очищает базу данных от всех данных
- d. Автоматически обновляет структуру таблиц

№ 10. Задание с единичным выбором. Выберите один правильный ответ.

Чем отличается тип данных JSONB от типа JSON в PostgreSQL?

- a. Различия между ними минимальны, выбор зависит лишь от предпочтений разработчика
- b. Тип JSON хранит данные в бинарном формате, тогда как JSONB — в текстовом
- c. Тип JSON позволяет индексировать содержимое документа, а JSONB — нет
- d. Тип JSONB оптимизирован для быстрого поиска и индексации, тогда как JSON предназначен исключительно для хранения данных

2.3.2. Практические задания для оценки компетенции «ПК-3 ППК-Р3.1»

№ 1. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1. Разработка структуры базы данных.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1. Разработка структуры базы данных

Необходимо

1. Создать таблицу для учета **пользователей** системы. Предусмотреть справочник военных **должностей**.
2. Изменить таблицу `measurement\_batch` изменить поле `username`.

Задание сдается каждым учащимся в собственной репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 2. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 3. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 4. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.3. Практические задания для оценки компетенции «ПК-4 ППК-Р4.2»

№ 5. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1. Разработка структуры базы данных.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1.

Разработка структуры базы данных

Необходимо

1. Создать таблицу для учета **пользователей** системы. Предусмотреть справочник военных **должностей**.

2. Изменить таблицу `measurement\_batch` изменить поле `username`.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 6. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4.

Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все** константы

Температура. Минимальное значение-58, максимальное 58, указывается в цельсиях Давления. Минимальное значение 500, максимальное 900, указывается в мм рт ст Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров

- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.

- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)

- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.

- Написать отдельный **pgSQL** скрипт, который сформирует тестовые данные. Нужно добавить несколько пользователей в таблицу employees, а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 7. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 8. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.4. Практические задания для оценки компетенции «ПК-5 ППК-Р5.1»

№ 9. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1. Разработка структуры базы данных.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1.

Разработка структуры базы данных

Необходимо

1. Создать таблицу для учета **пользователей** системы. Предусмотреть справочник военных **должностей**.

2. Изменить таблицу `measurement\_batch` изменить поле `username`.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 10. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.5. Практические задания для оценки компетенции «ПК-5 ППК-Р5.2»

№ 11. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1. Разработка структуры базы данных.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 1.

Разработка структуры базы данных

Необходимо

1. Создать таблицу для учета **пользователей** системы. Предусмотреть справочник военных **должностей**.

2. Изменить таблицу `measurement\_batch` изменить поле `username`.

Задание сдается каждым учащимся в собственной репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 12. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.6. Практические задания для оценки компетенции «ПК-2 ППК-Р2.2»

№ 13. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Создание скриптов

Необходимо:

- Создать pgSQL скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.

- Добавить в pgSQL скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.

- В скрипт включить создание связей по ключам между таблицами и тип данных interpolation

- Написать отдельный pgSQL скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственной репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 14. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции
 - Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL
 - Провести расчет от **0 - 40** градусов с шагом **0,01**
 - Замерить скорость расчета .
 - **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 15. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4.

Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все** константы

Температура. Минимальное значение-58, максимальное 58, указывается в цельсиях Давления. Минимальное значение 500, максимальное 900, указывается в мм рт ст Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров
- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.
- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)
- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.
- Написать отдельный **pgSQL** скрипт, который сформирует тестовые данные. Нужно добавить несколько пользователей в таблицу employees , а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 16. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5.

Использование массивов

- Создать и заполнить набор таблиц со справочной информации для расчета среднего отклонения воздуха (**Таблица 2**)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmDmk.md

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

для **Ветрового ружья** и **ДМК**

- Написать хранимую процедуру для расчета среднего отклонения воздуха. На вход передаем температуру, на выход получаем результат в виде массива.

- Дополнить "главный" скрипт. В него включить процедуру, таблицы, добавление данных и пример расчета по аналогии

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250214.sql

- Написать SQL запрос для формирования отчета вида:
ФИО | Должность | Кол-во измерений | Количество ошибочных данных |
Отсортировать по полю "Количество ошибочных данных"

- Сделать в виде **PullRequest** и в виде картинки прислать результат расчета среднего отклонения воздуха

№ 17. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7.

Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5-ти измерений

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 18. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)

- `Скорость среднего ветра` (Таблица 3)

- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же, в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 19. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 20. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.7. Практические задания для оценки компетенции «ПК-2 ППК-Р2.3»

№ 21. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Создание скриптов

Необходимо:

- Создать `pgSQL` скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.
- Добавить в `pgSQL` скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.
- В скрипт включить создание связей по ключам между таблицами и тип данных `interpolation`
- Написать отдельный `pgSQL` скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 22. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции
 - Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL
 - Провести расчет от **0 - 40** градусов с шагом **0,01**
 - Замерить скорость расчета .
 - **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 23. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4.

Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все константы**

Температура. Минимальное значение-58, максимальное 58, указывается в градусах Цельсия. Давление. Минимальное значение 500, максимальное 900, указывается в мм рт ст. Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров
- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.
- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)
- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.
- Написать отдельный `pgSQL` скрипт, который сформирует тестовые данные. Нужно

добавить несколько пользователей в таблицу employees , а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственной репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 24. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов

- Создать и заполнить набор таблиц со справочной информации для расчета среднего отклонения воздуха (**Таблица 2**)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmDmk.md

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

для **Ветрового ружья** и **ДМК**

- Написать хранимую процедуру для расчета среднего отклонения воздуха. На вход передаем температуру, на выход получаем результат в виде массива.

- Дополнить "главный" скрипт. В него включить процедуру, таблицы, добавление данных и пример расчета по аналогии

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250214.sql

- Написать SQL запрос для формирования отчета вида:
ФИО | Должность | Кол-во измерений | Количество ошибочных данных |
Отсортировать по полю "Количество ошибочных данных"

- Сделать в виде **PullRequest** и в виде картинки прислать результат расчета среднего отклонения воздуха

№ 25. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE
"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5- ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 26. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)
- `Скорость среднего ветра` (Таблица 3)
- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 27. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>
- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 28. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы
https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template
- Пример подключения Api
https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42
<https://docs.postgrest.org/en/v12/index.html>
- Примеры
http://localhost:3000/temp_input_params?order=id.desc&limit=1
- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.8. Практические задания для оценки компетенции «ПК-2 ППК-Р2.4»

№ 29. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов

Необходимо:

- Создать `pgSQL` скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.
- Добавить в `pgSQL` скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.
- В скрипт включить создание связей по ключам между таблицами и тип данных `interpolation`
- Написать отдельный `pgSQL` скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 30. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции
 - Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL
 - Провести расчет от **0 - 40** градусов с шагом **0,01**
 - Замерить скорость расчета .
 - **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 31. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все** константы

Температура. Минимальное значение-58, максимальное 58, указывается в градусах Цельсия. Минимальное значение 500, максимальное 900, указывается в мм рт.ст. Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров
- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.
- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)
- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.
- Написать отдельный **pgSQL** скрипт, который сформирует тестовые данные. Нужно добавить несколько пользователей в таблицу employees, а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 32. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов

- Создать и заполнить набор таблиц со справочной информацией для расчета среднего отклонения воздуха (**Таблица 2**)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmDmk.md

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

для **Ветрового ружья** и **ДМК**

- Написать хранимую процедуру для расчета среднего отклонения воздуха. На вход передаем температуру, на выход получаем результат в виде массива.
- Дополнить "главный" скрипт. В него включить процедуру, таблицы, добавление данных и пример расчета по аналогии

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250214.sql

- Написать SQL запрос для формирования отчета вида:
ФИО | Должность | Кол-во измерений | Количество ошибочных данных |
Отсортировать по полю "Количество ошибочных данных"

- Сделать в виде **PullRequest** и в виде картинки прислать результат расчета среднего отклонения воздуха

№ 33. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5- ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 34. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)
- `Скорость среднего ветра` (Таблица 3)
- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 35. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов

расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 36. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.9. Практические задания для оценки компетенции «ПК-4 ППК-Р4.1»

№ 37. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Создание скриптов

Необходимо:

- Создать pgSQL скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.

- Добавить в pgSQL скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.

- В скрипт включить создание связей по ключам между таблицами и тип данных interpolation

- Написать отдельный pgSQL скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 38. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 39. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4.

Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все** константы

Температура. Минимальное значение-58, максимальное 58, указывается в цельсиях Давления. Минимальное значение 500, максимальное 900, указывается в мм рт ст Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров

- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.

- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)

- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.

- Написать отдельный **pgSQL** скрипт, который сформирует тестовые данные. Нужно добавить несколько пользователей в таблицу employees , а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 40. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5.

Использование массивов

- Создать и заполнить набор таблиц со справочной информации для расчета среднего отклонения воздуха (**Таблица 2**)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmDmk.md

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

для **Ветрового ружья** и **ДМК**

- Написать хранимую процедуру для расчета среднего отклонения воздуха. На вход передаем температуру, на выход получаем результат в виде массива.

- Дополнить "главный" скрипт. В него включить процедуру, таблицы, добавление

данных и пример расчета по аналогии

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250214.sql

- Написать SQL запрос для формирования отчета вида:
ФИО | Должность | Кол-во измерений | Количество ошибочных данных |
Отсортировать по полю "Количество ошибочных данных"

- Сделать в виде **PullRequest** и в виде картинки прислать результат расчета среднего отклонения воздуха

№ 41. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 42. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5- ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 43. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)
- `Скорость среднего ветра` (Таблица 3)
- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 44. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>
- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: **input_params**, **employees** и т.д.

- Приложить изображение от Postman с результатами расчета
 - Отправлен POST запрос на добавление данных
 - Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 45. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42
<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.10. Практические задания для оценки компетенции «ПК-8 ППК-Р7.1»

№ 46. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Создание скриптов

Необходимо:

- Создать `pgSQL` скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.

- Добавить в `pgSQL` скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.

- В скрипт включить создание связей по ключам между таблицами и тип данных `interpolation`

- Написать отдельный `pgSQL` скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 47. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 48. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)

- `Скорость среднего ветра` (Таблица 3)

- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 49. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.11. Практические задания для оценки компетенции «ПК-8 ППК-Р7.3»

№ 50. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Создание скриптов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Создание скриптов

Необходимо:

- Создать pgSQL скрипт в который поместить прошлое задание по созданию инфраструктуры проекта.

- Добавить в pgSQL скрипт таблицу поправок по температуре (Таблица 1). Так же, добавить данные в эту таблицу.

- В скрипт включить создание связей по ключам между таблицами и тип данных interpolation

- Написать отдельный pgSQL скрипт для расчета поправки к температуре на основе Таблица 1 с применением интерполяции.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 51. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции
 - Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL
 - Провести расчет от **0 - 40** градусов с шагом **0,01**
 - Замерить скорость расчета .
 - **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 52. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета
- `Средние отклонения температуры воздуха` (Таблица 2)
- `Скорость среднего ветра` (Таблица 3)
- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 53. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.12. Практические задания для оценки компетенции «ПК-2 ППК-Р2.1»

№ 54. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтерн-

ативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 55. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4. Наполнение данными.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 4.

Наполнение данными

Необходимо:

- Создать таблицу с настройками для проверки входных данных **все** константы

Температура. Минимальное значение-58, максимальное 58, указывается в цельсияхД-авления. Минимальное значение500, максимальное 900, указывается в мм рт ст Направление ветра. Минимальное значение 0, максимальное значение 59 и т.д.

- Создать собственный тип данных для передачи входных параметров

- Написать собственную функцию на вход должны подаваться входные параметры, а на выходе собственный тип данных.

- Функция должна проверять входные параметры. При нарушении граничных параметров формировать [raise error](#)

- Допisać и оформить "основной" скрипт. В него включить все функции для расчета интерполяции и метео-средний.

- Написать отдельный **pgSQL** скрипт, который сформирует тестовые данные. Нужно добавить несколько пользователей в таблицу employees , а так же не менее 100 измерений с разными параметрами для каждого пользователя. Для указания высоты, давления, температуры использовать функцию random(). Для создания скрипта использовать циклы.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 56. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5. Использование массивов.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 5.

Использование массивов

- Создать и заполнить набор таблиц со справочной информации для расчета среднего отклонения воздуха (**Таблица 2**)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmDmk.md

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

для **Ветрового ружья** и **ДМК**

- Написать хранимую процедуру для расчета среднего отклонения воздуха. На вход передаем температуру, на выход получаем результат в виде массива.

- Дополнить "главный" скрипт. В него включить процедуру, таблицы, добавление данных и пример расчета по аналогии

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250214.sql

- Написать SQL запрос для формирования отчета вида:
ФИО | Должность | Кол-во измерений | Количество ошибочных данных |
Отсортировать по полю "Количество ошибочных данных"

- Сделать в виде **PullRequest** и в виде картинки прислать результат расчета среднего отклонения воздуха

№ 57. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)
- `Скорость среднего ветра` (Таблица 3)
- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 58. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 59. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.13. Практические задания для оценки компетенции «ПК-6 ИП-2.1»

№ 60. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 61. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 62. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgorithmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5-ти измерений

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 63. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 64. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.14. Практические задания для оценки компетенции «ПК-6 ИП-2.2»

№ 65. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 66. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 67. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7.

Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра**

(таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5-ти измерений

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 68. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 69. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.15. Практические задания для оценки компетенции «ПК-8 ППК-Р7.2»

№ 70. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 71. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 8. Расчеты.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 8. Расчеты

- Необходимо написать хранимые процедуры (можно одну универсальную) для расчета

- `Средние отклонения температуры воздуха` (Таблица 2)

- `Скорость среднего ветра` (Таблица 3)

- `Направление среднего ветра` (Таблица 3)

Для **Ветрового ружья** и **ДМК**

- Текст хранимой процедуры включить в общий скрипт. Там же ,в общем скрипте, вызвать процедуры и сделать тестовый расчет

Пример

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/scripts/HomeWork20250228.sql

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 72. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42
<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.16. Практические задания для оценки компетенции «ПК-8 ППК-Р7.4»

№ 73. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 3. Альтернативный расчет.

Домашнее задание

Необходимо

- Реализовать алгоритм расчета интерполяции

- Написать приложение на любом языке программирования с использованием собственной базы данных на PostgreSQL

- Провести расчет от **0 - 40** градусов с шагом **0,01**

- Замерить скорость расчета .

- **Запрещается** использовать кеширование в приложении.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 74. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.17. Практические задания для оценки компетенции «ПК-3 ППК-Р3.2»

№ 75. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 76. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>
- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 77. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.18. Практические задания для оценки компетенции «ПК-4 ППК-Р4.3»

№ 78. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов

интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 79. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.19. Практические задания для оценки компетенции «ПК-5 ППК-Р5.3»

№ 80. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 81. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.20. Практические задания для оценки компетенции «ПК-7 ППК-Р6.2»

№ 82. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 83. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7.

Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5- ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 84. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную

таблицу `temp_input_params` для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: `input_params`, `employees` и т.д.

- Приложить изображение от Postman с результатами расчета

— Отправлен POST запрос на добавление данных

— Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 85. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.21. Практические задания для оценки компетенции «ПК-7 ППК-Р6.3»

№ 86. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6. Использование процедур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 6.

Использование процедур

- Необходимо разработать отдельную **хранимую** процедуру для расчет всех вариантов интерполяции.

- Разработать приложение на любом языке программирования. Для запуска расчета использовать хранимую процедуру. Приложение должно запустить хранимую процедуру и получить результаты расчетов интерполяции

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13459>

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 87. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7.

Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием CTE

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5-ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 88. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.22. Практические задания для оценки компетенции «ПК-7 ППК-Р6.1»

№ 89. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7. Разработка универсальных структур.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 7.

Разработка универсальных структур

Необходимо:

- Реализовать структуры для хранения корректировок для **расчета среднего ветра** (таблица 3)

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Docs/AlgoritmBp.md

2 В общий скрипт добавить индексы

4 Запрос с прошлого задания оформить в виде представления (View и CTE)

<https://bki.forlabs.ru/app/my-courses/1035/tasks/13518>

5 Написать новый отчет и оформить его в виде View и использованием СТЕ

"Самая эффективная высота измерения"

| ФИО пользователя | Звание | Мин. высота метеопоста | Макс. высота метепоста | Всего измерений | Из них ошибочны |

Необходимо вывести список пользователей у которых меньше 10 ошибок в наборе данных, которые делали не менее 5-ти измерения

Ошибку определять обязательно с использованием **функции проверки корректности**.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задания

№ 90. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 9. Триггеры.

Кейс «Разработка ПАК "Артиллерист"».

Самостоятельная работа 9. Триггеры

Необходимо:

- Добавить в **основной** скрипт триггерную функцию и триггер. Добавить временную таблицу **temp_input_params** для взаимодействия с ResApi Подключить триггер к данной таблице и подключить RestApi.

Примеры на занятии:

- <https://docs.postgrest.org/en/v12/index.html>

- https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L41

- Доработать **триггерную** функцию. Добавить в нее расчет. Далее, сохранение результатов расчета. В случае успешного расчета, то копирование данных в основные таблицы: input_params, employees и т.д.

- Приложить изображение от Postman с результатами расчета

 - Отправлен POST запрос на добавление данных

 - Отправлен GET запрос на получение данных. Последняя добавленная запись из таблицы. Результат расчета.

Задание сдается каждым учащимся в собственном репозитории (Git) в виде ссылки на PullRequest. Далее, производится проверка задан

№ 91. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.23. Практические задания для оценки компетенции «ПК-5 ППК-Р5.4»

№ 92. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.24. Практические задания для оценки компетенции «ПК-7 ППК-Р6.4»

№ 93. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 10. Сборка приложения.

Кейс «Разработка ПАК "Артиллерист"».

Практическое занятие 10. Сборка приложения

Необходимо подключить RestApi, Postgres к фронту

- Шаблон Html страницы

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Template

- Пример подключения Api

https://git.hostfl.ru/VolovikovAlex/Study2025/src/master/_Infra/docker-compose.yaml#L42

<https://docs.postgrest.org/en/v12/index.html>

- Примеры

http://localhost:3000/temp_input_params?order=id.desc&limit=1

- Подключить данные полученные в результате расчетов и заполнить таблицу с историей

2.3.25. Проекты для оценки компетенции «ПК-5 ППК-Р5.1»

№ 94. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Подготовка рабочего места для разработки систем на основе PostgreSQL.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.

Подготовка рабочего места для разработки систем на основе PostgreSQL

Цель задания

Подготовить и задокументировать воспроизводимое рабочее окружение для разработки и тестирования систем на основе PostgreSQL с использованием Linux, Docker и PostgREST, а также сформировать навыки настройки и проверки инфраструктуры, приближённой к production-среде.

❓ Исходные условия

- Операционная система: **Linux**
 - Инструменты контейнеризации: **Docker, Docker Compose**
 - СУБД: **PostgreSQL**
 - REST-слой над БД: **PostgREST**
-

❓ Задание

1. Подготовка окружения

1. Установить и настроить:
 - Linux (нативно или в виртуальной среде);
 - Docker и Docker Compose.
 2. Проверить корректность установки:
 - запуск контейнера PostgreSQL;
 - доступ к БД из контейнера и с хост-машины.
-

2. Развёртывание PostgreSQL в Docker

1. Создать `docker-compose.yml`, обеспечивающий:
 - запуск PostgreSQL с персистентным хранилищем данных;
 - задание параметров БД через переменные окружения;
 - открытый порт для подключения.
 2. Выполнить начальную инициализацию БД:
 - создание схемы;
 - создание ролей (минимум: `admin` и `app-user`);
 - настройка прав доступа.
-

3. Развёртывание PostgREST

1. Добавить сервис **PostgREST** в `docker-compose.yml`.
2. Настроить:
 - подключение PostgREST к PostgreSQL;
 - схему и роль, от имени которой выполняются запросы;
 - базовую конфигурацию безопасности.
3. Проверить:
 - доступность REST API;
 - корректность выполнения CRUD-операций.

4. Проверка и тестирование

1. Создать тестовую таблицу и заполнить её данными.
 2. Выполнить:
 - запрос к БД напрямую (psql);
 - запрос к данным через PostgREST (curl / HTTP-клиент).
 3. Зафиксировать результаты проверок.
-

5. Документирование

Подготовить краткую техническую документацию, включающую:

- структуру проекта;
 - инструкцию по запуску (`docker compose up`);
 - описание ролей и схем;
 - примеры запросов к API;
 - возможные проблемы и способы их устранения.
-

🔗 Формат сдачи

- Репозиторий (Git):
 - `docker-compose.yml`;
 - конфигурационные файлы PostgREST;
 - SQL-скрипты инициализации;
 - `README.md`.
 - Все команды должны быть воспроизводимы на «чистой» машине.
-

🔗 Критерии оценки

- корректность и воспроизводимость окружения;
 - безопасность базовой конфигурации;
 - работоспособность PostgreSQL и PostgREST;
 - качество документации;
 - инженерная аккуратность и структура проекта.
-

🔗 Бонус (по желанию)

- использование `.env` для конфигурации;
 - `healthcheck` для сервисов;
 - базовая настройка RLS;
 - автоматическая инициализация данных.
-

2.3.26. Проекты для оценки компетенции «ПК-5 ППК-Р5.2»

№ 95. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Подготовка рабочего места для разработки систем на основе PostgreSQL.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2. Подготовка рабочего места для разработки систем на основе PostgreSQL

🔗 Цель задания

Подготовить и задокументировать воспроизводимое рабочее окружение для разработки и тестирования систем на основе PostgreSQL с использованием Linux, Docker и PostgREST, а также сформировать навыки настройки и проверки инфраструктуры, приближённой к production-среде.

🔗 Исходные условия

- Операционная система: **Linux**
- Инструменты контейнеризации: **Docker, Docker Compose**
- СУБД: **PostgreSQL**
- REST-слой над БД: **PostgREST**

🔗 Задание

1. Подготовка окружения

1. Установить и настроить:
 - Linux (нативно или в виртуальной среде);
 - Docker и Docker Compose.
2. Проверить корректность установки:
 - запуск контейнера PostgreSQL;
 - доступ к БД из контейнера и с хост-машины.

2. Развёртывание PostgreSQL в Docker

1. Создать `docker-compose.yml`, обеспечивающий:
 - запуск PostgreSQL с персистентным хранилищем данных;
 - задание параметров БД через переменные окружения;
 - открытый порт для подключения.
2. Выполнить начальную инициализацию БД:
 - создание схемы;
 - создание ролей (минимум: `admin` и `app-user`);
 - настройка прав доступа.

3. Развёртывание PostgREST

1. Добавить сервис **PostgREST** в `docker-compose.yml`.
2. Настроить:
 - подключение PostgREST к PostgreSQL;
 - схему и роль, от имени которой выполняются запросы;
 - базовую конфигурацию безопасности.
3. Проверить:
 - доступность REST API;
 - корректность выполнения CRUD-операций.

4. Проверка и тестирование

1. Создать тестовую таблицу и наполнить её данными.
2. Выполнить:
 - запрос к БД напрямую (psql);
 - запрос к данным через PostgREST (curl / HTTP-клиент).
3. Зафиксировать результаты проверок.

5. Документирование

Подготовить краткую техническую документацию, включающую:

- структуру проекта;
- инструкцию по запуску (`docker compose up`);
- описание ролей и схем;
- примеры запросов к API;
- возможные проблемы и способы их устранения.

❓ Формат сдачи

- Репозиторий (Git):
 - `docker-compose.yml`;
 - конфигурационные файлы PostgREST;
 - SQL-скрипты инициализации;
 - `README.md`.
 - Все команды должны быть воспроизводимы на «чистой» машине.
-

❓ Критерии оценки

- корректность и воспроизводимость окружения;
 - безопасность базовой конфигурации;
 - работоспособность PostgreSQL и PostgREST;
 - качество документации;
 - инженерная аккуратность и структура проекта.
-

❓ Бонус (по желанию)

- использование `.env` для конфигурации;
- `healthcheck` для сервисов;
- базовая настройка RLS;
- автоматическая инициализация данных.

2.3.27. Проекты для оценки компетенции «ПК-7 ППК-Р6.4»

№ 96. Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.
Подготовка рабочего места для разработки систем на основе PostgreSQL.

Кейс «Разработка ПАК "Артиллерист"». Самостоятельная работа 2.
Подготовка рабочего места для разработки систем на основе PostgreSQL

❓ Цель задания

Подготовить и задокументировать воспроизводимое рабочее окружение для разработки и тестирования систем на основе PostgreSQL с использованием Linux, Docker и PostgREST, а также сформировать навыки настройки и проверки инфраструктуры, приближённой к production-среде.

❓ Исходные условия

- Операционная система: **Linux**
 - Инструменты контейнеризации: **Docker, Docker Compose**
 - СУБД: **PostgreSQL**
 - REST-слой над БД: **PostgREST**
-

❓ Задание

1. Подготовка окружения

1. Установить и настроить:

- Linux (нативно или в виртуальной среде);
- Docker и Docker Compose.

2. Проверить корректность установки:

- запуск контейнера PostgreSQL;
- доступ к БД из контейнера и с хост-машины.

2. Развёртывание PostgreSQL в Docker

1. Создать `docker-compose.yml`, обеспечивающий:
 - запуск PostgreSQL с персистентным хранилищем данных;
 - задание параметров БД через переменные окружения;
 - открытый порт для подключения.
2. Выполнить начальную инициализацию БД:
 - создание схемы;
 - создание ролей (минимум: `admin` и `app-user`);
 - настройка прав доступа.

3. Развёртывание PostgREST

1. Добавить сервис **PostgREST** в `docker-compose.yml`.
2. Настроить:
 - подключение PostgREST к PostgreSQL;
 - схему и роль, от имени которой выполняются запросы;
 - базовую конфигурацию безопасности.
3. Проверить:
 - доступность REST API;
 - корректность выполнения CRUD-операций.

4. Проверка и тестирование

1. Создать тестовую таблицу и наполнить её данными.
2. Выполнить:
 - запрос к БД напрямую (`psql`);
 - запрос к данным через PostgREST (`curl` / HTTP-клиент).
3. Зафиксировать результаты проверок.

5. Документирование

- Подготовить краткую техническую документацию, включающую:
- структуру проекта;
 - инструкцию по запуску (`docker compose up`);
 - описание ролей и схем;
 - примеры запросов к API;

— возможные проблемы и способы их устранения.

❓ Формат сдачи

— Репозиторий (Git):

—

`docker-compose.yml`;

— конфигурационные файлы PostgREST;

— SQL-скрипты инициализации;

— `README.md`.

— Все команды должны быть воспроизводимы на «чистой» машине.

❓ Критерии оценки

— корректность и воспроизводимость окружения;

— безопасность базовой конфигурации;

— работоспособность PostgreSQL и PostgREST;

— качество документации;

— инженерная аккуратность и структура проекта.

❓ Бонус (по желанию)

— использование `.env` для конфигурации;

— `healthcheck` для сервисов;

— базовая настройка RLS;

— автоматическая инициализация данных.

3. Промежуточная аттестация

3.1. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и опыта деятельности

Зачет является заключительным этапом процесса формирования компетенций обучающегося при изучении дисциплины и имеет целью проверку и оценку знаний обучающегося по теории, и применению полученных знаний, умений и навыков при решении практических задач.

Зачет проводится по расписанию, сформированному учебно-методическим управлением, в сроки, предусмотренные календарным учебным графиком. Зачет принимается преподавателем, ведущим лекционные занятия.

Зачет проводится только при предъявлении обучающимся зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой дисциплины. Обучающимся на зачете представляется право выбрать один из билетов. Время подготовки к ответу составляет 30 минут. По истечении установленного времени обучающийся должен ответить на вопросы билета. Результаты зачета оцениваются по четырехбалльной системе и заносятся в зачетно-экзаменационную ведомость и зачетную книжку. В зачетную книжку заносятся только положительные оценки. Подписанный преподавателем экземпляр ведомости сдаётся не позднее следующего дня в деканат.

В случае неявки обучающегося на зачет в зачетно-экзаменационную ведомость делается отметка «не явка». Обучающиеся, не прошедшие промежуточную аттестацию по дисциплине, должны ликвидировать академическую задолженность в установленном локальными нормативными актами порядке.

3.2. Вопросы к зачету с оценкой

№	Вопрос	Код компетенции
1.	Архитектура PostgreSQL: основные процессы и их взаимодействие. MVCC в PostgreSQL: принципы работы, преимущества и ограничения. WAL: назначение, устройство и роль в обеспечении надёжности. Механизмы блокировок и конкуренции транзакций. Уровни изоляции транзакций в PostgreSQL и их практические последствия. Управление памятью: shared buffers, work_mem, maintenance_work_mem.	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3
2.	В рамках разработки базы данных для ПАК «Артиллерист» необходимо автоматизировать создание структуры базы данных средствами SQL-скриптов. Опишите назначение и структуру сценария развертывания базы данных. Какие объекты должны быть созданы с помощью скриптов (схемы, таблицы, ограничения, индексы, начальные данные и др.), почему важно обеспечить возможность повторного выполнения скриптов, и какие преимущества дает использование SQL-скриптов при командной разработке и сопровождении проекта?	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3
3.	В рамках разработки ПАК «Артиллерист» необходимо реализовать алгоритм интерполяции для определения промежуточных значений баллистических параметров по табличным данным. Объясните назначение интерполяции в системе автоматизации расчетных задач поддержки артиллерии, опишите алгоритм линейной интерполяции, приведите его математическую основу и предложите способ реализации алгоритма средствами PostgreSQL (SQL-функции или PL/pgSQL). Обоснуйте требования к структуре таблиц и обеспечению корректности вычислений.	ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.4, ПК-4 ППК-Р4.1, ПК-8 ППК-Р7.1, ПК-8 ППК-Р7.2
4.	В рамках разработки ПАК «Артиллерист» необходимо подготовить SQL-скрипт на языке PL/pgSQL, предназначенный для автоматического формирования тестовых данных базы данных. Скрипт должен создавать несколько записей в таблице employees, а затем для каждого пользователя автоматически генерировать не менее 100 записей с результатами измерений. Значения высоты, давления и температуры должны формироваться с использованием функции random(). При реализации скрипта необходимо использовать циклы PL/pgSQL, обеспечив автоматическое заполнение базы данных без ручного добавления записей. Скрипт должен быть готов к многократному запуску и использоваться для тестирования и проверки работы программного комплекса.	ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.4, ПК-4 ППК-Р4.1, ПК-8 ППК-Р7.1, ПК-8 ППК-Р7.2, ПК-8 ППК-Р7.3

№	Вопрос	Код компетенции
5.	Необходимо разработать отдельную хранимую процедуру на PL/pgSQL, предназначенную для расчета всех предусмотренных вариантов интерполяции в рамках базы данных ПАК «Артиллерист». Процедура должна принимать необходимые исходные параметры, обращаться к табличным данным, выполнять расчет промежуточных значений и возвращать результат интерполяции в удобном для дальнейшей обработки виде. При разработке необходимо предусмотреть обработку некорректных или отсутствующих данных, а также обеспечить возможность повторного использования процедуры в других расчетных сценариях программного комплекса.	ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-2 ППК-Р2.4, ПК-3 ППК-Р3.1, ПК-3 ППК-Р3.2, ПК-4 ППК-Р4.1, ПК-6 ИП-2.2
6.	Необходимо разработать хранимые процедуры на PL/pgSQL для выполнения расчетов по табличным данным в рамках кейса «Разработка ПАК "Артиллерист"». Допускается реализация как нескольких отдельных процедур, так и одной универсальной хранимой процедуры, позволяющей выполнять разные типы расчетов. Процедура должна обеспечивать расчет следующих параметров: средние отклонения температуры воздуха по данным Таблицы 2, скорость среднего ветра по данным Таблицы 3 и направление среднего ветра по данным Таблицы 3. При реализации необходимо предусмотреть получение исходных данных из соответствующих таблиц базы данных, выполнение необходимых вычислений, возврат результата расчета и обработку ситуаций, когда данные отсутствуют или заданы некорректно.	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-2 ППК-Р2.4, ПК-4 ППК-Р4.1
7.	Архитектура PostgreSQL: основные процессы и их взаимодействие. MVCC в PostgreSQL: принципы работы, преимущества и ограничения. WAL: назначение, устройство и роль в обеспечении надёжности. Механизмы блокировок и конкуренции транзакций. Уровни изоляции транзакций в PostgreSQL и их практические последствия. Управление памятью: shared buffers, work_mem, maintenance_work_mem.	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3
8.	Проектирование схем данных под высоконагруженные системы. Партиционирование: виды, сценарии применения, ограничения. Стратегии работы с историческими и временными данными (temporal data). Подходы к реализации multitenancy в PostgreSQL. Денормализация: когда оправдана и какие риски несёт.	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3
9.	Индексы в PostgreSQL: виды, выбор и влияние на производительность. EXPLAIN и EXPLAIN ANALYZE: интерпретация планов выполнения. Использование JSONB: преимущества, ограничения, индексация. Триггеры и функции: границы ответственности между БД и приложением. Расширения PostgreSQL: примеры и области применения (PostGIS, FDW и др.).	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3

3.3. Тематика курсовых работ

По данной дисциплине выполнение курсовых проектов (работ) не предусматривается.

3.4. Материалы для компьютерного тестирования обучающихся

Общие критерии оценивания

Процент правильных ответов	Оценка
91% – 100%	5 (отлично)
81% – 90%	4 (хорошо)
71% – 80%	3 (удовлетворительно)
Менее 70%	2 (неудовлетворительно)

Соответствие вопросов теста индикаторам формируемых и оцениваемых компетенций

№ вопроса в тесте	Код индикатора компетенции
1	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
2	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
3	ПК-4 ППК-Р4.1
4	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
5	ПК-4 ППК-Р4.1
6	ПК-4 ППК-Р4.1
7	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4
8	ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2
9	ПК-4 ППК-Р4.1
10	ПК-2 ППК-Р2.1, ПК-2 ППК-Р2.2, ПК-2 ППК-Р2.3, ПК-3 ППК-Р3.1, ПК-4 ППК-Р4.1, ПК-4 ППК-Р4.2, ПК-4 ППК-Р4.3, ПК-5 ППК-Р5.1, ПК-6 ИП-2.1, ПК-6 ИП-2.2, ПК-7 ППК-Р6.2, ПК-8 ППК-Р7.1, ПК-2 ППК-Р2.4

Ключ ответов

№ вопроса в тесте	Номер ответа (или ответ, или соответствие)
1	b
2	d
3	d
4	c
5	a

№ вопроса в тесте	Номер ответа (или ответ, или соответствие)
6	d
7	c
8	c
9	a
10	d

Перечень тестовых вопросов

№ 1. Задание с единичным выбором. Выберите один правильный ответ.

Что такое PostgreSQL?

- a. Фреймворк для веб-разработки
- b. Система управления реляционными базами данных
- c. Инструмент для анализа больших данных
- d. Язык программирования

№ 2. Задание с единичным выбором. Выберите один правильный ответ.

Что вернёт следующий SQL-запрос в PostgreSQL?

- a. Все записи из таблицы сотрудников (employees)
- b. Все записи из таблицы отделов (departments)
- c. Только общие поля обеих таблиц
- d. Произведение каждой строки из таблицы сотрудников на каждую строку из таблицы

отделов

№ 3. Задание с единичным выбором. Выберите один правильный ответ.

Когда выполняется триггер AFTER INSERT в PostgreSQL?

- a. Перед вставкой новой записи в таблицу
- b. После неудачной попытки вставки новой записи
- c. Во время проверки ограничений целостности перед вставкой
- d. После успешной вставки новой записи в таблицу

№ 4. Задание с единичным выбором. Выберите один правильный ответ.

Как правильно объявляется локальная переменная внутри блока PL/pgSQL в PostgreSQL?

- a. VAR <переменная> := <тип>;
- b. <переменная> AS <тип>;
- c. DECLARE <переменная> <тип>;
- d. DECLARE <переменная> TYPE <тип>;

№ 5. Задание с единичным выбором. Выберите один правильный ответ.

Что обозначают аббревиатуры DML в контексте PostgreSQL?

- a. Data Manipulation Language (язык манипулирования данными)
- b. Database Management Layer (уровень управления базой данных)
- c. Dynamic Metadata Library (динамическая библиотека метаданных)
- d. Distributed Multi-level Locking (распределённая многоуровневая блокировка)

№ 6. Задание с единичным выбором. Выберите один правильный ответ.

Какая последовательность команд в PostgreSQL запускает транзакцию, записывает изменения и подтверждает их выполнение?

- a. DECLARE → SELECT → END
- b. SAVEPOINT → DELETE → RELEASE
- c. START TRANSACTION → INSERT → ROLLBACK
- d. BEGIN TRANSACTION → UPDATE → COMMIT

№ 7. Задание с единственным выбором. Выберите один правильный ответ.

Как называется специальная структура, используемая для передачи параметров внутри процедуры в PostgreSQL?

- a. Cursor
- b. Argument
- c. Parameter
- d. Trigger

№ 8. Задание с единственным выбором. Выберите один правильный ответ.

Какая конструкция SQL-запроса позволяет создать столбец массива строковых значений в таблице PostgreSQL?

- a. column_name STRING_ARRAY
- b. column_name ARRAY(VARCHAR)
- c. column_name VARCHAR[]
- d. column_name []VARCHAR

№ 9. Задание с единственным выбором. Выберите один правильный ответ.

Что означает кнопка «Vacuum Analyze» в интерфейсе pgAdmin?

- a. Запускает процедуру оптимизации и сбора статистики для улучшения производительности запросов
- b. Выполняет резервное копирование всей базы данных
- c. Полностью очищает базу данных от всех данных
- d. Автоматически обновляет структуру таблиц

№ 10. Задание с единственным выбором. Выберите один правильный ответ.

Чем отличается тип данных JSONB от типа JSON в PostgreSQL?

- a. Различия между ними минимальны, выбор зависит лишь от предпочтений разработчика
- b. Тип JSON хранит данные в бинарном формате, тогда как JSONB — в текстовом
- c. Тип JSON позволяет индексировать содержимое документа, а JSONB — нет
- d. Тип JSONB оптимизирован для быстрого поиска и индексации, тогда как JSON предназначен исключительно для хранения данных