Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.01.ДВ.02.01 Анализ больших данных
(индекс, наименование дисциплины в соответствии с учебным планом)

09.03.03 Прикладная информатика
(код, наименование направления подготовки/специальности)

Прикладная информатика в управлении корпоративными информационными системами
(наименование образовательной программы)

Бакалавр
квалификация

Очная форма обучения
(форма обучения)

Год набора – 2026
Донецк

Автор(ы)-составитель(и) РПД:

Литвак Елена Геннадиевна, канд. экон. наук, доцент кафедры информационных технологий

Заведующий кафедрой:

Брадул Наталья Валерьевна, канд. физ.-мат. наук, заведующий кафедрой информационных технологий

Рабочая программа дисциплины Б1.В.01.ДВ.02.01 Анализ больших данных одобрена на заседании кафедры информационных технологий факультета государственной службы и управления Донецкого филиала РАНХиГС.

Протокол № 7 от «05» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.В.01.ДВ.02.01 Анализ больших данных обеспечивает формирование у обучающихся следующих общепрофессиональных компетенций*:

ОТФ /ТФ и реквизиты ПС *(при наличии)* **	**Код компетенции ****	**Наименование Компетенции ****	**Код индикатора достижения компетенций ****	**Наименование индикатора достижения компетенций ****	**Образовательный результат ****
-	ПК-1.	Способность адаптировать бизнес-процессы заказчика ИС к возможностям типовой ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС	ПК-1.1	Собирает исходные данные у заказчика ИС о его бизнес-процессах	ПК-1.1. З-2 Знает Источники информации, необходимой для профессиональной деятельности при выполнении работ по созданию (модификации) и сопровождению ИС; ПК-1.1. У-2 Умеет Анализировать исходную документацию в рамках выполнения работ по созданию (модификации) и сопровождению ИС

** Дисциплина может формировать компетенцию полностью или частично.*
***Должно соответствовать Приложению 1 к образовательной программе*

2. Объем и место дисциплины в структуре образовательной программы

Общий объем дисциплины:

2,00 з.е., 72 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий: 30 ак. час на контактную работу с преподавателем, из них 10 ак. час на лекции и 20 ак. час на практические занятия. 38 ак. час на самостоятельную работу обучающихся.

Б1.В.01.ДВ.02.01 Анализ больших данных реализуется на 8-м семестре 4-го курса после изучения дисциплин:

- Разведывательный анализ данных.
- Базы данных.

.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очнаяформа обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час												Форма текущего контроля успеваемости, промежуточной аттестации
			Контактная работа обучающихся с преподавателем по видам учебных занятий									Самостоятельная работа			
			Период теоретического обучения						Период промежуточной аттестации (сессия)						
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Каттэк	Контроль	СРкр	СРэк	СР	
			Л	ВЛ	ЛР	ПЗ									
РАЗДЕЛ 1. ОТ PANDAS К МАСШТАБИРУЕМЫМ РЕШЕНИЯМ															
Тема 1	Когда данные перестают быть «маленькими»: ограничения Pandas	9	1	0	0	2	0	0	0	0		0	0	6	Контрольные вопросы, практические занятия, КТ1
Тема 2	Dask: первые шаги	9	1	0	0	2	0	0	0	0	0	0	0	6	Контрольные вопросы,

															практические занятия, КТ 1
Тема 3	Архитектура PySpark и модель MapReduce	10	2	0	0	2	0	0	0	0	0	0	0	6	Контрольные вопросы, практические занятия, КТ 1
РАЗДЕЛ 2. ПРОМЫШЛЕННАЯ АНАЛИТИКА НА PYSPARK															
Тема 4	Spark SQL и оптимизация запросов	12	2	0	0	4	0	0	0	0	0	0	0	6	Контрольные вопросы, практические занятия, КТ 2
Тема 5	Шуффлинг и партиционирование	12	2	0	0	4	0	0	0	0	0	0	0	6	Контрольные вопросы, практические занятия, КТ 2
Тема 6	Инструментарий аналитика: оконные функции и агрегации	14	2	0	0	4	0	0	0	0	0	0	0	8	Контрольные вопросы, практические занятия, КТ 2
Промежуточная аттестация		4	0	0	0	0	0	0		4	0		0	0	Зачет с оценкой
Итого		**72**	**10**	**0**	**0**	**20**	**0**	**0**	**0**	**4**	**0**		**0**	**38**	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).
ИК – индивидуальные консультации.
КСР – контроль самостоятельной работы
КЭ – консультации перед экзаменом
Каттэк – контактная работа на аттестацию в период экзаменационных сессий
Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения
СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.
СРэк – самостоятельная работа на подготовку к экзамену.
СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

3.2. Содержание дисциплины

РАЗДЕЛ 1. ОТ PANDAS К МАСШТАБИРУЕМЫМ РЕШЕНИЯМ

Тема 1. Когда данные перестают быть «маленькими»: ограничения Pandas. ПК-1.1

Признаки того, что Pandas перестает справляться: ошибки памяти, замедление swap, время выполнения более 10 минут. Примеры из реальных задач (логи за месяц, данные датчиков). Правило «50% RAM» как порог перехода. Демонстрация падения Pandas на 20 ГБ данных при 16 ГБ RAM.

Тема 2. Dask: первые шаги. ПК-1.1

Краткое содержание: Что такое ленивые вычисления. Чанки и партиции. Граф задач. Ключевое отличие от Pandas: ничего не вычисляется без `.compute()`. Синтаксис, идентичный Pandas для 90% операций. Параллелизм на всех ядрах процессора без изменения кода.

Тема 3. Архитектура PySpark и модель MapReduce. ПК-1.1

Краткое содержание: Почему Spark - индустриальный стандарт. Архитектура driver/executors. RDD и DataFrame. MapReduce простыми словами: map (разбить) → shuffle (перегруппировать) → reduce (собрать). Ленивые трансформации и action'ы (`.show()`, `.collect()`, `.count()`). Сравнение Dask (одна машина) и Spark (кластер).

РАЗДЕЛ 2. ПРОМЫШЛЕННАЯ АНАЛИТИКА НА PYSPARK

Тема 4. Spark SQL и оптимизация запросов. ПК-1.1

Краткое содержание: Почему DataFrame API удобнее RDD. Регистрация временных представлений (temp views). Использование чистого SQL внутри Spark: `spark.sql("SELECT ...")`. Catalyst Optimizer. Просмотр физического плана через `.explain()`. Эвристики оптимизации: фильтровать рано, выбирать нужные колонки, избегать cartesian join.

Тема 5. Шуффлинг и партиционирование. ПК-1.1

Краткое содержание: Что происходит при `groupBy`, `join`, `orderBy`. Почему shuffle - самая медленная операция. Репартиционирование: `repartition()` (полный shuffle) vs `coalesce()` (уменьшение без shuffle). Признаки проблем: спарклайны с shuffle, раздутые executor'ы. Правило: держать партиции размером 100–200 МБ.

Тема 6. Инструментарий аналитика: оконные функции и агрегации. ПК-1.1

Краткое содержание: Оконные функции в Spark SQL: `row_number()`, `rank()`, `lag()`, `lead()`. Оконные агрегации (`sum() over partition by`). Применение к реальным задачам: поиск топ-3 пользователей по группе, расчет скользящего среднего, детектирование аномалий по времени.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.В.01.ДВ.02.01 Анализ больших данных входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа – это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).	Ответ считается верным, если правильно указаны цифры или буквы

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)

Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).	Ответ считается верным, если правильно указана вся последовательность цифр
Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).	Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа

Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ	Ответ считается верным: 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5. Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС Донецкого филиала РАНХиГС.

Итоговая балльная оценка	**Традиционная система**	**Бинарная система**	**ECTS**	
			Для традиционной системы	**Для бинарной системы**
90-100	Отлично	Зачтено	A	P/ Passed
80-89	Хорошо		B	P/ Passed
75-79			C	P/ Passed
70-74	Удовлетворительно		B	P/ Passed
60-69			E	P/ Passed
0-59	Неудовлетворительно	Не зачтено	F	F/Failed

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
100 баллов	100 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.В.01.ДВ.02.01 Анализ больших данных используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

Контрольные вопросы для проведения опроса, задания открытого типа на практических занятиях, контрольные задания

Таблица 5.1.

Распределение баллов по видам учебной деятельности (БРС)

Раздел/Темы	Формы текущего контроля		КТ
	УО	ПЗ	
Р-1. / Т-1	5	5	20
Р-1. / Т-2	5	5	
Р-1. / Т-3	5	5	
Р-2. / Т-4	5	5	20
Р-2. / Т-5	5	5	
Р-2. / Т-6	5	5	
Итого: 100 б	30	30	40

УО – устный опрос;
ТЗ – тестовое задание;
КЗ – контрольные задания;
ПЗ – практическое занятие;
Д – доклад;
КТ – контрольные точки.

Критерии оценивания опроса:

Балы	Описание критерия
4-5	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
2-3	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
1	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

0* - в журнал академической группы не выставляется

Критерии оценивания практических занятий:

Балы	Описание критерия	
4-5	Свыше 90% правильных ответов.	Обучающийся демонстрирует глубокое познание в освоенном материале.
2-3	Свыше 70% правильных ответов.	Обучающимся материал освоен полностью, без существенных ошибок.
1	Реализовано более 50% поставленных задач	Обучающимся материал освоен не полностью, имеются значительные пробелы в знаниях.

0	Реализовано менее 30% поставленных задач.	Обучающимся материал не освоен, знания обучающегося ниже базового уровня.

0* - в журнал академической группы не выставляется

Критерии оценивания контрольных заданий:

Балы	Описание критерия
18-20	Обучающимся задание выполнено без ошибок и в полном объеме.
15-17	Обучающимся в целом задание выполнено, имеются отдельные неточности или недостаточно полные ответы, не содержащие ошибок.
10-14	Обучающимся допущены отдельные ошибки при выполнении задания
0-9	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

0* - в журнал академической группы не выставляется

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):

РАЗДЕЛ 1. ОТ PANDAS К МАСШТАБИРУЕМЫМ РЕШЕНИЯМ

Тема 1. Когда данные перестают быть «маленькими»: ограничения Pandas

Контрольные вопросы:

1. Какие три признака указывают на то, что Pandas перестаёт справляться с объёмом данных?
2. Почему увеличение оперативной памяти не всегда решает проблему больших данных в Pandas?
3. Что произойдёт с Pandas, если загрузить 20 ГБ данных на машину с 16 ГБ RAM и включенным swap?
4. Назовите два типа операций в Pandas, которые особенно чувствительны к объёму данных.

Практическое задание:

Напишите скрипт на Pandas, который генерирует DataFrame из 10 млн строк (поля: `user_id`, `amount`, `timestamp`), выполняет группировку по `user_id` со средним `amount`, и замеряет время выполнения и использование памяти. Затем увеличьте объём до 50 млн строк. Зафиксируйте момент, когда скрипт перестаёт выполняться или падает по памяти. Сделайте вывод.

Тема 2. Dask: первые шаги

Контрольные вопросы:

1. Что такое «ленивые вычисления» в Dask и как они отличаются от немедленных вычислений в Pandas?
2. Зачем нужен метод .compute() в Dask?
3. Что такое граф задач (task graph) и как он связан с партициями данных?
4. Почему Dask может обрабатывать данные, не влезающие в оперативную память, а Pandas - нет?

Практическое задание:

Возьмите код из предыдущего задания (группировка 50 млн строк). Перепишите его с использованием Dask (`dask.dataframe`). Добейтесь успешного выполнения. Сравните время выполнения и сложность кода. Добавьте этап репартиционирования (`.repartition(npartitions=8)`) и замерьте разницу.

Тема 3. Архитектура PySpark и модель MapReduce

Контрольные вопросы:

1. Из каких основных компонентов состоит архитектура Spark (driver, executors)?
2. В чём разница между трансформацией (transformation) и действием (action) в Spark? Приведите по два примера.
3. Объясните модель MapReduce на простом примере: подсчёт слов в тексте.
4. Почему в Spark используются RDD и DataFrame, а не обычные Python-списки?

Практическое задание:

Напишите PySpark-скрипт, который читает текстовый файл (10 ГБ синтетических логов), считает частоту HTTP-методов (GET, POST, PUT, DELETE) и выводит результат. Намеренно используйте `.collect()` для получения результата, затем замените на `.take(10)` и объясните разницу в поведении.

РАЗДЕЛ 2. ПРОМЫШЛЕННАЯ АНАЛИТИКА НА PYSPARK

Тема 4. Spark SQL и оптимизация запросов

Контрольные вопросы:

1. Как зарегистрировать временное представление (temp view) в Spark и выполнить SQL-запрос?
2. Что делает Catalyst Optimizer и зачем он нужен?
3. Как посмотреть физический план выполнения запроса в Spark?
4. Назовите два правила оптимизации запросов, которые Spark применяет автоматически.

Практическое задание:

Загрузите датасет в PySpark. Реализуйте один и тот же аналитический запрос двумя способами: через DataFrame API и через Spark SQL. Сравните планы выполнения через `.explain()`. Убедитесь, что Catalyst Optimizer сгенерировал одинаковые планы. Добавьте фильтр на раннем этапе и покажите разницу в плане.

Тема 5. Шуффлинг и партиционирование

Контрольные вопросы:

5. Что такое shuffle в Spark и почему он считается медленной операцией?
6. В чём разница между `repartition(n)` и `coalesce(n)`?
7. Какой рекомендуемый размер партиции в Spark (в мегабайтах) и почему?
8. Как узнать, что ваш запрос вызвал shuffle (по экрану или по плану)?

Практическое задание:

Дан запрос, выполняющий `groupBy` и `join` двух таблиц по 5 ГБ каждая. Исходно запрос работает 15 минут. Используя `.explain()` и изменение числа партиций, добейтесь ускорения до 5 минут. Напишите, какие изменения вы сделали и почему они сработали.

Тема 2.3. Инструментарий аналитика: оконные функции и агрегации

Контрольные вопросы:

1. Что такое оконная функция в Spark SQL? Приведите пример использования `row_number()`.
2. Чем оконная агрегация отличается от обычной `GROUP BY`?
3. Как с помощью оконных функций найти топ-3 записи в каждой группе?

4. Какие оконные функции можно использовать для анализа временных рядов (lag, lead)?

Практическое задание

Дан датасет транзакций (`user_id`, `transaction_date`, `amount`). Напишите Spark-запрос, который для каждого пользователя вычисляет:

накопленную сумму трат по дням (оконная сумма)

топ-2 самых дорогих транзакции (row_number)

разницу в сумме между текущей и предыдущей транзакцией (lag)

Результат выведите в виде таблицы.

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точкой по разделу (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной работы	Максимальное количество баллов за работу в рамках КР, которое может набрать студент	Коэффициент веса контрольной работы	Результат контрольной работы, участвующий в формировании итоговой балльной оценки по дисциплине
КТ 1	100	0,2	20
КТ 2	100	0,2	20
Итого:	x	0,4	40

Формула расчета результата контрольной работы:

Результат контрольной работы = Количество баллов за точку в рамках КТ X Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

КТ 1

Задача 1.

Напишите код на Python, который:

Генерирует Pandas DataFrame из 10 млн строк с полями: `id`, `category` (5 категорий), `value` (случайное число)

Выполняет группировку по `category` со средним `value`

Замеряет время выполнения и память (можете использовать `time` и `memory_profiler`)

Увеличивает количество строк до 50 млн и фиксирует ошибку/падение

В комментариях объясните, почему произошла ошибка

Задача 2.

Дан фрагмент кода на Pandas, который падает на 20 ГБ данных:

```
python
```

```python
import pandas as pd
df = pd.read_csv('huge_logs.csv')
result = df[df['status'] == 404].groupby('ip').size()
print(result)
```

Перепишите этот код на Dask так, чтобы он работал на 20 ГБ данных. Объясните, зачем нужен `.compute()`, и предложите, как можно оптимизировать запрос (например, репартиционирование).

Задача 3

Представьте, что у вас есть PySpark DataFrame `logs` с полями: `user_id`, `action`, `timestamp`. Напишите цепочку трансформаций, которая:

фильтрует записи с `action = 'purchase'`

группирует по `user_id`

считает количество покупок на пользователя

выводит топ-10 пользователей

В ответе укажите:

Код на PySpark

Какие операции в вашем коде - трансформации, а какие - действия

Что произойдёт, если после всех трансформаций добавить `.collect()`?

КТ 2

Задача 1.

Дан датасет `sales` с полями: `region`, `product`, `amount`, `date`.
Напишите два эквивалентных запроса, которые вычисляют суммарные продажи по регионам за январь 2024 года:

один - с использованием DataFrame API

второй - с использованием Spark SQL

В ответе укажите:

Оба варианта кода

Как вы убедитесь, что оба запроса дадут одинаковый результат?

Какой вариант вы предпочтёте и почему?

Задача 2.

Дан медленный запрос:

```python
df1 = spark.read.parquet("table1")  # 10 ГБ, 200 партиций
df2 = spark.read.parquet("table2")  # 8 ГБ, 200 партиций

result = df1.join(df2, "id").filter(df1.status == "active").count()
```

Запрос работает 15 минут. Ваша задача:

Определить, какая операция вызывает shuffle (и почему)

Предложить три конкретных изменения для ускорения запроса

Написать оптимизированную версию кода

Задача 3.

Дан датасет `user_actions` с полями: `user_id`, `action_date`, `amount`.
Напишите PySpark-запрос, который для каждого пользователя:

Вычисляет накопленную сумму `amount` по датам (оконная сумма)

Ранжирует действия по убыванию `amount` внутри каждого пользователя (window + `row_number`)

Оставляет только топ-2 самых больших `amount` для каждого пользователя. Результат должен содержать поля: `user_id`, `action_date`, `amount`, `cumulative_sum`, `rank_within_user`.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация проводится в форме *зачета* с оценкой в 8-м семестре в письменной форме. Обучающийся получает три теоретических вопроса и одно практическое задание.

6.2. Типовые оценочные материалы промежуточной аттестации.

РАЗДЕЛ 1. ОТ PANDAS К МАСШТАБИРУЕМЫМ РЕШЕНИЯМ

Тема 1. Когда данные перестают быть «маленькими»: ограничения Pandas.

Вопросы к зачёту:

1. Объясните «правило 50% RAM» при работе с Pandas. Почему даже если данные занимают 8 ГБ, а памяти 16 ГБ, Pandas всё равно может упасть? Какие факторы, кроме размера самих данных, влияют на потребление памяти?
2. Опишите три стратегии работы с данными, которые не влезают в память, не используя Dask или Spark (только средства Python и Pandas). В чём недостатки каждой стратегии?

Практическое задание:

Вам дан CSV-файл объёмом 8 ГБ (логи с полями: `timestamp`, `user_id`, `event_type`, `value`). На машине 8 ГБ RAM. Напишите код на чистом Pandas (без Dask/Spark), который:

Прочитает файл **по частям** (chunks)
Для каждой части вычислит среднее `value` по `event_type`
Агрегирует промежуточные результаты в общий результат
Выведет итоговую таблицу

Дополнительное требование: код не должен использовать более 2 ГБ памяти в любой момент времени.

Тема 2. Dask: первые шаги

Вопросы к зачёту:

В чём принципиальное отличие Dask от Pandas с точки зрения выполнения операций? Что такое «граф задач» (task graph) и как он связан с

ленивыми вычислениями?

Объясните, почему Dask может обработать 50 ГБ данных на ноутбуке с 16 ГБ RAM, а Pandas - нет. Какая цена за эту возможность (компромиссы Dask)?

Что произойдёт, если в Dask вызвать `.compute()` на слишком больших промежуточных данных? Как можно контролировать использование памяти при выполнении?

Практическое задание:

Напишите скрипт на Dask, который:

Загружает 15 ГБ данных (многократное чтение одного файла или синтетическая генерация)

Выполняет последовательность операций: фильтр (`value > 100`), группировку по `category`, расчёт среднего и стандартного отклонения

Сохраняет результат в Parquet

Добавляет визуализацию графа задач (`.visualize()`) и сохраняет её как PNG

Дополнительное требование: продемонстрируйте, как изменится граф задач, если добавить `.repartition(4)` перед группировкой.

Тема 3. Архитектура PySpark и модель MapReduce

Вопросы к зачёту:

1. Нарисуйте (или опишите словами) архитектуру Spark-приложения: driver, executors, кластер-менеджер. Что происходит, когда вы вызываете `.count()` на датафрейме?
2. Объясните модель MapReduce на примере задачи: «Дан лог посещений сайта (user_id, url, timestamp). Найти топ-10 пользователей по количеству уникальных URL». Разбейте на стадии map, shuffle, reduce.
3. В чём разница между `df.collect()` и `df.take(10)`? Почему `collect()` считается опасным на больших данных? Что произойдёт, если вызвать `collect()` на датафрейме размером 100 ГБ?

Практическое задание:

Напишите PySpark-скрипт, который:

Создаёт RDD из 100 млн случайных чисел (или читает большой файл)

Реализует **ручную** MapReduce (через RDD API, без DataFrame) для подсчёта, сколько чисел попадает в каждый из 10 интервалов: [0-0.1), [0.1-0.2), ..., [0.9-1.0]

Перепишите то же самое с использованием DataFrame API и `groupBy`

Сравните время выполнения обоих подходов

Дополнительное требование: в коде должны быть комментарии, поясняющие, где происходит map, shuffle, reduce.

РАЗДЕЛ 2. ПРОМЫШЛЕННАЯ АНАЛИТИКА НА PYSPARK

Тема 4. Spark SQL и оптимизация запросов

Вопросы к зачёту:

1. Что такое Catalyst Optimizer? Приведите два конкретных примера оптимизаций, которые Spark выполняет автоматически. Как посмотреть, был ли применён конкретный оптимизатор?
2. Когда имеет смысл использовать Spark SQL вместо DataFrame API? В чём преимущества и недостатки каждого подхода? Как зарегистрировать временную таблицу и выполнить SQL-запрос?

Практическое задание:

Вам дан запрос на DataFrame API, который работает медленно:

```python
result = df \
    .select("user_id", "action", "amount") \
    .filter(df.action == "purchase") \
    .groupBy("user_id") \
    .sum("amount") \
    .orderBy("sum(amount)", ascending=False)
```

Выполните следующие действия:

Напишите **эквивалентный запрос на Spark SQL**

Для **обоих** запросов выведите физический план выполнения (`.explain()`)

Определите, есть ли в планах shuffle (и на каком этапе)

Предложите оптимизацию (одно изменение) и покажите новый план

Тема 5. Шуффлинг и партиционирование

Вопросы к зачёту:

1. Что такое shuffle в Spark? Почему он считается самой дорогой операцией? Опишите, что физически происходит с данными во время shuffle (на уровне партиций и узлов).
2. В чём разница между `repartition()` и `coalesce()`? В каком сценарии безопасно использовать `coalesce()`, а в каком он не поможет или навредит?
3. Как выбрать оптимальное количество партиций? Какая формула рекомендуется (например, число партиций относительно числа ядер и

размера данных)? Что произойдёт, если партиций слишком много или слишком мало?

Практическое задание:

Дан датасет (10 ГБ, 200 партиций по умолчанию). Выполняется join двух таблиц по ключу:

```python
df1.join(df2, "user_id", "inner")
```

Join работает 20 минут. Ваша задача:

Напишите код, который замеряет время выполнения исходного join'a

Измените количество партиций для **обоих** датафреймов перед join'ом (предложите два варианта: через `repartition` и через `coalesce`)

Сравните время выполнения трёх вариантов (исходный, с repartition, с coalesce)

Добавьте broadcast hint, если одна из таблиц маленькая (< 2 ГБ), и снова замерьте

Тема 6. Инструментарий аналитика: оконные функции и агрегации

Вопросы к зачёту:

1. Что такое оконные функции в Spark? Приведите три примера оконных функций (агрегационные, ранжирующие, функции смещения). Чем оконная функция отличается от `GROUP BY`?
2. Как определить окно в PySpark? Что означают параметры `partitionBy`, `orderBy`, `rowsBetween`? Приведите пример окна для расчёта скользящего среднего за последние 3 записи.

Практическое задание:

Дан датасет `sales` (10 ГБ) с полями: `store_id`, `date`, `daily_sales`. Напишите PySpark-запрос, который вычисляет:

Для каждого магазина - **процент** дневных продаж от общих продаж этого магазина (оконная сумма с partitionBy store_id)

Для каждого магазина - разницу между продажами текущего дня и предыдущего (`lag`)

Ранжирует дни по объёму продаж внутри каждого магазина (`dense_rank`)

Оставляет только топ-3 дня по продажам для каждого магазина

Результат должен содержать колонки: `store_id`, `date`, `daily_sales`, `pct_of_total`, `diff_from_prev`, `rank_by_sales`, и быть отфильтрован по `rank_by_sales <= 3`.

Дополнительное требование: укажите в комментариях, какая

операция в вашем запросе вызывает shuffle и почему.

6.3. Критерии и шкала оценивания на основе БРС.

Соответствие государственной шкалы оценивания академической успеваемости и шкалы ECTS при экзамене

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий

Компьютер с операционной системой RedOS, на котором установлены Apache, PHP, Mysql, интерпретатор Python, VSCode (или другой редактор).

7. Методические материалы по освоению дисциплины

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Студенты могут установить диалог с преподавателем, получать консультации по выполнению заданий. В качестве оценочных средств на протяжении семестра используются практические задания.

Обучение по дисциплине «Анализ больших данных» предполагает изучение курса на аудиторных занятиях (лекции, практические занятия) и самостоятельную работу студентов. Практические занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения студент должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

– знакомит с новым учебным материалом;
– разъясняет учебные элементы, трудные для понимания;
– систематизирует учебный материал;
– ориентирует в учебном процессе.

Работа обучающегося на лекции:

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность обучающегося. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим обучающимся.

Подготовка к практическим занятиям:

Подготовку к каждому практическому занятию каждый обучающийся должен начать с ознакомления с планом, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции. Все новые понятия по изучаемой теме необходимо внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура практического занятия:

В зависимости от содержания и количества отведенного времени на изучение каждой темы может практическое занятие состоять из четырех-пяти частей:

1. Устный опрос.

2. Обсуждение теоретических вопросов, определенных программой дисциплины.

3. Выполнение практических заданий с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома.

4. Подведение итогов занятия.

Работа с литературными источниками:

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на занятиях, выявить широкий спектр мнений по изучаемой проблеме.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Лебедев, А. С. Методы Big Data : учебно-методическое пособие / А. С. Лебедев, Ш. Г. Магомедов. — Москва : РТУ МИРЭА, 2021. — 91 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/182452 (дата обращения: 12.05.2026). — Режим доступа: для авториз. пользователей.
2. Янссенс, Й. Python Polars. Подробное руководство : руководство / Й. Янссенс, Т. Ньюдорп ; пер. с англ. А. Ю. Гинько. — Москва : ДМК Пресс, 2025. — 502 с. — ISBN 978-6-01140-650-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/514969 (дата обращения: 12.05.2026). — Режим доступа: для авториз. пользователей.

8.2. Дополнительная литература

3. Кудрявцева, И. Г. Основы бизнес-аналитики : учебно-методическое пособие / И. Г. Кудрявцева. — Москва : РТУ МИРЭА, 2025. — 237 с. — ISBN 978-5-7339-2548-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/498062 (дата обращения: 12.05.2026). — Режим доступа: для авториз. пользователей.
4. Груздев, А. В. Предварительная подготовка данных в Python / А. В. Груздев. — Москва : ДМК Пресс, 2023 — Том 2 : План, примеры и метрики качества — 2023. — 814 с. — ISBN 978-5-93700-177-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/314948 (дата обращения: 12.05.2026). — Режим доступа: для авториз. пользователей.

8.4 Интернет-ресурсы

1. Информационно-правовой портал ГАРАНТ.РУ. – URL: https://www.garant.ru/
2. Научная электронная библиотека eLIBRARY.RU. – URL: https://elibrary.ru/
3. Научная электронная библиотека «КиберЛенинка». – URL: https://cyberleninka.ru
4. Электронно-библиотечная система «Лань». – URL: http://e.lanbook.com
5. База знаний по ОС RedOS – URL: https://redos.red-soft.ru/base/
6. Документация по Mysql – URL: https://metanit.com/sql/mysql/

7. Документация по Python – URL: https://www.python.org/

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Материально-техническое обеспечение дисциплины включает в себя:

- лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном;
- помещения для проведения практических занятий, оборудованные учебной мебелью.

Дисциплина поддержана соответствующими программными продуктами с открытой лицензией: RedOS, MariaDB, Apache, интерпретатор Python, phpMyAdmin.

Вуз обеспечивает каждого обучающегося рабочим местом в компьютерном классе в соответствии с объемом изучаемых дисциплин, обеспечивает выход в сеть Интернет.

Помещения для самостоятельной работы обучающихся включают следующую оснащенность: столы аудиторные, стулья, доски аудиторные, компьютеры с подключением к локальной сети института (для компьютерных аудиторий) и Интернет. Для изучения учебной дисциплины используются автоматизированная библиотечная информационная система и электронные библиотечные системы.