Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Врио заместителя начальника колледжа
по учебно-методической работе
М.Ю. Никишин

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00861698655D171818D836B833A86412BD
Владелец: НИКИШИН МИХАИЛ ЮРЬЕВИЧ
Действителен: с 13.02.2026 до 09.05.2027

Фонд оценочных средств

(приложение к рабочей программе профессионального модуля)

ПМ.02 АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ

основной профессиональной образовательной программы среднего профессионального образования по специальности

09.02.12 Техническая эксплуатация и сопровождение информационных систем

МО-09 02 12-ПМ.02.ФОС

РАЗРАБОТЧИК	Иванова Е.Д.
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ	Судьбина Н.А.
ГОД РАЗРАБОТКИ	2026

СОДЕРЖАНИЕ

1 Паспорт фонда оценочных средств 3
1.1 Область применения фонда оценочных средств 3
1.2 Результаты освоения учебной дисциплины 3
2 Перечень оценочных средств и критерии оценивания 5
3 Оценочные средства текущего контроля и промежуточной аттестации 8
4 Сведения о фонде оценочных средств и его согласование 20

1 Паспорт фонда оценочных средств

1.1 Область применения фонда оценочных средств

Фонд оценочных средств предназначен для оценки результатов освоения учебной дисциплины ПМ.02 Администрирование баз данных

1.2 Результаты освоения учебной дисциплины

В результате контроля и оценки по учебной дисциплине осуществляется комплексная проверка освоения следующих общих компетенций:

ПК 2.1. Устанавливать и сопрягать компоненты СУБД

ПК 2.2. Выполнять администрирование баз данных

ПК 2.3. Управлять доступом к базам данных

ПК 2.4. Выполнять резервное копирование и восстановление баз данных

ПК 2.5. Проводить мониторинг и журналирование событий

ПК 2.6. Обеспечивать безопасную работу сервера СУБД

Таблица 1- Планируемые результаты обучения

Код формируемых компетенции	**Планируемые результаты**	
	Общие	**Дисциплинарные (предметные)**
ПК2.1. Устанавливать и сопрягать компоненты СУБД	Иметь практический опыт: - развертывания и инсталляции серверов СУБД на различных операционных системах; - интеграции СУБД с внешними сервисами и приложениями. Уметь: - выбирать оптимальную аппаратную конфигурацию; - сопрягать СУБД с серверами приложений и веб-серверами. Знать: - архитектуру СУБД и системные требования к серверному оборудованию.	- Уметь устанавливать СУБД (PostgreSQL, MS SQL, MySQL) в консольном и графическом режимах на Windows/Linux сервера; - Уметь настраивать сетевые параметры СУБД для обеспечения удаленных клиентских подключений (конфигурация сокетов, портов, слушающих интерфейсов); - Настраивать драйверы и библиотеки сопряжения (ODBC, JDBC, ADO.NET) для интеграции СУБД с прикладным ПО.
ПК 2.2. Выполнять администрирование баз данных	Иметь практический опыт: -управления конфигурационными файлами, дисковым пространством и системными каталогами СУБД. Уметь: - изменять параметры работы сервера без прерывания обслуживания; - переносить файлы БД на новые физические носители. Знать:	- Уметь управлять основными файлами конфигурации СУБД (например, postgresql.conf, my.cnf); - Владеть механизмами управления табличными пространствами (Tablespaces) для эффективного распределения таблиц по разным дисковым массивам; - Уметь производить плановое обслуживание баз данных: очистку устаревших версий строк, дефрагментацию и пересчет статистики индексов.

Код формируемых компетенции	Планируемые результаты	
	Общие	**Дисциплинарные (предметные)**
	- структуру хранения данных на диске, внутренние системные таблицы СУБД.	
ПК 2.3. Управлять доступом к базам данных	Иметь практический опыт: - проектирования и реализации ролевых моделей разграничения прав доступа на предприятии. Уметь: - создавать учетные записи пользователей, групп и ролей; - гибко распределять привилегии и запреты. Знать: - механизмы аутентификации и авторизации в современных СУБД.	- Уметь настраивать методы проверки подлинности пользователей (от md5/scram-sha-256 до интеграции с Active Directory/LDAP); - Писать и выполнять SQL-скрипты управления правами (GRANT, REVOKE, DENY) на уровне серверов, баз, схем и конкретных объектов (таблиц, процедур); - Уметь настраивать контекстную безопасность и ограничение доступа на уровне отдельных строк (Row-Level Security — RLS).
ПК 2.4. Выполнять резервное копирование и восстановление баз данных	Иметь практический опыт: - разработки регламентов бэкапов и ликвидации последствий техногенных аварий. Уметь: - создавать резервные копии всех типов; - проводить процедуру восстановления данных с минимальным простоем. Знать: - стратегии резервного копирования и виды отказов систем..	- Уметь создавать полные, дифференциальные/инкрементные бэкапы и архивировать журналы транзакций (WAL) с помощью CLI-утилит (например, pg_dump, pg_basebackup); - Безошибочно выполнять аварийное восстановление баз данных из бэкапов и файлов предзаписи на заданный момент времени в прошлом (технология PITR); - Настраивать планировщики задач (cron, SQL Server Agent) для автоматизации процесса создания резервных копий по расписанию.
ПК 2.5. Проводить мониторинг и журналирование событий	Иметь практический опыт: - анализа логов, отслеживания системных метрик производительности сервера. Уметь: - использовать встроенные системные представления мониторинга; - выявлять медленные операции и блокировки. Знать: - структуру журналов событий СУБД, способы трассировки запросов	- Уметь анализировать файлы журналов ошибок и отладки СУБД для локализации причин аварийного завершения работы; - Активно применять системные динамические представления (pg_stat_activity, sys.dm_exec_requests) для выявления зависших сессий и взаимных блокировок (Deadlocks); - Настраивать и анализировать лог медленных запросов (Slow Query Log) для сбора аналитики о производительности.- Знать требования законодательства РФ (в частности, ФЗ-152 «О персональных данных») к защите информационных систем и баз данных.
ПК2.6.Обеспечивать безопасную работу сервера СУБД	Иметь практический опыт: - защиты серверов баз данных от внешних и внутренних ИБ-угроз. Уметь:	Уметь настраивать шифрование сетевого трафика между клиентом и СУБД по протоколу SSL/TLS; - Уметь внедрять политики сквозного аудита действий суперпользователей и фиксации несанкционированных попыток входа;

Код формируемых компетенции	Планируемые результаты	
	Общие	**Дисциплинарные (предметные)**
	- настраивать сквозной аудит безопасности; - включать шифрование трафика и данных. Знать: - методы защиты от сетевых атак, инъекций кода, требования законов о защите данных.	- Настраивать ограничения безопасности на уровне операционной системы (права на папки с данными, изоляция системного пользователя СУБД).

2 Перечень оценочных средств и критерии оценивания

2.2 К оценочным средствам текущего контроля успеваемости относятся:

- контрольные вопросы к темам практических занятий.

2.3 К оценочным средствам для промежуточной аттестации относятся:

- тестовые задания открытого и закрытого типа;
- билеты для экзамена.

2.4 Критерии оценки результатов освоения учебной дисциплины

Критерии оценивания теоретических знаний:

«Отлично» — выставляется обучающемуся, который:

Точно и исчерпывающе отвечает на теоретические вопросы билета в области проектирования, администрирования и защиты баз данных.

Демонстрирует безупречное владение ИТ-терминологией и точные формулировки понятий (СУБД, транзакции, нормализация, репликация, SQL-инъекции, ACID).

Показывает глубокое понимание архитектуры данных, аргументированно обосновывает выбор конкретной модели или СУБД под бизнес-задачи.

Свободно классифицирует и дифференцирует понятия (SQL/NoSQL, DDL/DML, кластеризованные/некластеризованные индексы).

Уверенно и аргументированно отвечает на дополнительные вопросы преподавателя.

Излагает материал технически грамотно, логично и последовательно.

«Хорошо» — выставляется обучающемуся, если его ответ удовлетворяет требованиям к отметке «отлично», но:

Допускаются единичные технические неточности или мелкие ошибки в формулировках стандартов SQL / принципов администрирования.

Недочеты исправляются обучающимся самостоятельно после наводящего вопроса преподавателя.

«Удовлетворительно» — выставляется обучающемуся, который показывает базовое понимание теории, но:

Неуверенно отвечает на вопросы билета, путает этапы проектирования или администрирования СУБД.

Допускает неточности в ключевых понятиях (правила нормализации, уровни изолированности транзакций).

Затрудняется обосновать предложенную архитектуру БД или выбранный метод бэкапа.

Путает технические функции схожих объектов (триггеры и хранимые процедуры, типы резервного копирования).

Делает ошибки при ответах на дополнительные вопросы или излагает материал несвязно, с нарушением логики процессов.

«Неудовлетворительно» — выставляется обучающемуся, если он:

Демонстрирует полное отсутствие знаний по теоретическим основам модуля.

Не знает базовых терминов (база данных, первичный ключ, таблица).

Не может ответить на вопросы билета и дополнительные вопросы преподавателя.

Критерии оценивания практических умений:

«Отлично» — ставится, если обучающийся, который:

Самостоятельно и в полном объеме выполняет все пункты практического задания в среде СУБД.

Проектирует корректную архитектуру БД (ER-диаграмму), полностью устраняя аномалии и избыточность методом нормализации.

Пишет синтаксически верный, структурированный SQL-код, который выполняется сервером с первого раза без ошибок.

Безошибочно реализует задачи администрирования: настраивает ролевую модель (GRANT/REVOKE) и планы резервного копирования.

Аргументированно обосновывает логику работы написанных скриптов, триггеров, процедур и выбор типов данных.

Уверенно владеет интерфейсом СУБД и CLI-утилитами, оперативно решает практические микро-задачи.

«Хорошо» — выставляется обучающемуся, если задания выполнены на высоком уровне, но:

Допущены единичные синтаксические ошибки или опечатки в SQL, которые обучающийся сам находит по логам СУБД и исправляет.

При проектировании допущены мелкие погрешности (недочеты в CHECK-ограничениях, неоптимальные типы данных), не нарушающие целостность БД.

SQL-запросы работоспособны, но содержат избыточные конструкции (например, подзапросы вместо JOIN).

Настройка бэкапов и безопасности требует незначительных наводящих указаний преподавателя.

«Удовлетворительно» — выставляется обучающемуся, если он:

Выполняет практическое задание не в полном объеме (менее 70% от ТЗ) или со значительными задержками.

Допускает грубые ошибки в проектировании связей (PK/FK), приводящие к нарушению ссылочной целостности.

Пишет SQL-скрипты с логическими ошибками, затрудняется самостоятельно провести отладку кода.

Испытывает серьезные затруднения в администрировании СУБД (управление правами, параметры бэкапа).

Не может объяснить логику работы собственного кода, написанного с помощью подсказок.

«Неудовлетворительно» — выставляется обучающемуся, если:

Задание не выполнено или выполнено менее чем на 50%.

База данных не создана или полностью неработоспособна.

SQL-код содержит неустранимые ошибки, навыки проектирования и администрирования отсутствуют.

Критерии текущего контроля:

Шкала перевода баллов в отметки по пятибалльной системе

Отметка	**Число баллов, необходимое для получения отметки**
5 (отлично)	от 81-100 %
4 (хорошо)	от 61-80 %
3 (удовлетворительно)	от 41-60%
2 (неудовлетворительно)	менее 40 %

3 Оценочные средства текущего контроля и промежуточной аттестации

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ: ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5, ПК 2.6.

Контрольные вопросы к практическим занятиям

Практическое занятие 1. Каковы ключевые отличия между концептуальной, логической и физической моделями данных?

Концептуальная модель — это высокоуровневое бизнес-описание предметной области, не зависящее от конкретной СУБД. Она содержит только основные сущности и связи между ними (описывается понятным для заказчика языком).

Логическая модель — это детализация концептуальной модели. Здесь появляются все атрибуты, определяются первичные и внешние ключи, типы связей, а данные приводятся к нормальным формам. Она создается под определенный тип СУБД (например, реляционный), но все еще абстрагирована от конкретной программы.

Физическая модель — это точная копия будущей базы данных в конкретной СУБД (например, PostgreSQL). В ней указаны специфичные для этой СУБД типы данных (например, SERIAL, UUID), прописаны физические имена таблиц, ограничения, индексы и правила распределения памяти.

Практическое занятие 2. В чем заключается разница между связями типа «один-ко-многим» (1:M) и «многие-ко-многим» (M:N) при физической реализации таблиц?

Связь «один-ко-многим» (1:M) реализуется напрямую между двумя таблицами. Для этого первичный ключ (PRIMARY KEY) родительской таблицы («один») добавляется в дочернюю таблицу («многие») в качестве внешнего ключа (FOREIGN KEY). Дополнительных таблиц не требуется.

Связь «многие-ко-многим» (M:N) не может быть реализована напрямую в реляционной физической модели. Она требует создания третьей — связующей (промежуточной) таблицы. В эту таблицу помещаются как минимум два поля: внешние ключи, ссылающиеся на первичные ключи обеих исходных таблиц. Вместе они чаще всего образуют составной первичный ключ связующей таблицы.

Практическое занятие 3. Каковы условия приведения таблицы к третьей нормальной форме (3NF) и какие аномалии данных при этом устраняются?

Условия приведения к 3NF:

1. Таблица должна находиться во второй нормальной форме (2NF) (все не ключевые поля зависят от полного первичного ключа, а не от его части).

2. В таблице должны отсутствовать транзитивные зависимости — ни одно не ключевое поле не должно зависеть от другого не ключевого поля (все не ключевые поля зависят только от первичного ключа).

Устраняемые аномалии:

Аномалия вставки (INSERT): невозможность добавить информацию об одном объекте, пока не создана запись о другом (например, нельзя добавить отдел ИТ, пока в него не нанят хотя бы один сотрудник).

Аномалия удаления (DELETE): потеря важных независимых данных при удалении связанной записи (например, при увольнении последнего сотрудника стирается вся информация о самом существовании отдела).

Аномалия обновления (UPDATE): избыточное дублирование данных приводит к тому, что при изменении одного значения (например, названия города) приходится менять его в тысячах строк, что порождает риск несоответствия данных.

Практическое занятие 4. В чем разница между операторами категорий DDL (Data Definition Language) и DML (Data Manipulation Language)? Приведите примеры.

DDL (Язык определения данных) используется для создания, изменения и удаления самой структуры базы данных (ее объектов: таблиц, индексов, схем, представлений). Примеры операторов: CREATE, ALTER, DROP.

DML (Язык манипулирования данными) используется для непосредственной работы со строками данных внутри уже созданных таблиц (выборка, добавление, изменение и удаление записей). Примеры операторов: SELECT, INSERT, UPDATE, DELETE.

Практическое занятие 5. Для чего используются ограничения целостности PRIMARY KEY, FOREIGN KEY, UNIQUE и CHECK при создании таблиц?

PRIMARY KEY (Первичный ключ) — однозначно идентифицирует каждую строку в таблице. Запрещает дублирование значений и пустые поля (NOT NULL), гарантируя уникальность записей.

FOREIGN KEY (Внешний ключ) — связывает таблицы между собой. Он указывает на первичный ключ другой таблицы, гарантируя ссылочную целостность (нельзя добавить дочернюю запись без родительской или удалить родительскую, если на нее кто-то ссылается).

UNIQUE (Уникальность) — запрещает дублирование значений в столбце (например, для email или ИНН), но в отличие от первичного ключа позволяет хранить значения NULL.

CHECK (Проверка) — накладывает логическое условие на значения в столбце (например, CHECK (Age >= 18) или CHECK (Price > 0)), блокируя запись данных, не подходящих под критерий.

Практическое занятие 6. Каковы отличия в работе операторов соединения таблиц INNER JOIN, LEFT JOIN и FULL OUTER JOIN?

INNER JOIN (Внутреннее соединение) — возвращает только те строки, для которых нашлись совпадения (строгие соответствия ключей) в обеих таблицах. Если в одной из таблиц данных по ключу нет, строка отбрасывается.

LEFT JOIN (Левое внешнее соединение) — возвращает все строки из левой (первой) таблицы и только совпавшие строки из правой. Если для строки из левой таблицы нет соответствия в правой, поля правой таблицы заполняются значениями NULL.

FULL OUTER JOIN (Полное внешнее соединение) — объединяет результаты левого и правого соединений. Возвращает абсолютно все строки из обеих таблиц. Если совпадения по ключу нет, пустые места с любой стороны заполняются значениями NULL.

Практическое занятие 7. Для чего используются агрегатные функции в связке с операторами GROUP BY и HAVING? В чем отличие HAVING от WHERE?

Назначение: Агрегатные функции (SUM, AVG, COUNT, MIN, MAX) используются для проведения математических расчетов над множеством строк. Оператор GROUP BY разбивает строки на группы по определенному признаку, чтобы применить к ним эти функции (например, посчитать среднюю зарплату по каждому отделу).

Отличие HAVING от WHERE:

Оператор WHERE фильтрует исходные строки до того, как они будут сгруппированы. Он не может работать с результатами агрегатных функций (запрос типа WHERE SUM(price) > 100 вызовет ошибку).

Оператор HAVING фильтрует уже готовые сгруппированные результаты после отработки GROUP BY и может использовать агрегатные функции (например: GROUP BY department HAVING AVG(salary) > 50000).

Практическое занятие 8. В чем заключаются основные архитектурные отличия хранимой процедуры от пользовательской функции в СУБД?

Возвращаемое значение: Функция обязана возвращать результат (скалярное значение, массив или таблицу). Процедура не обязана ничего возвращать (выполняется как набор команд).

Управление транзакциями: Внутри хранимой процедуры можно управлять транзакциями (вызывать команды COMMIT и ROLLBACK). Внутри функции управлять транзакциями запрещено.

Контекст вызова: Функцию можно вызывать внутри обычных SQL-запросов (например: SELECT my_func(price) FROM products). Процедуру нельзя вызвать внутри SELECT, она запускается обособленно специальной командой (CALL procedure_name() в PostgreSQL или EXEC procedure_name в T-SQL).

Практическое занятие 9. Для каких целей создаются триггеры и чем отличаются триггеры типа BEFORE (или INSTEAD OF) от триггеров типа AFTER?

Цель создания: Триггеры нужны для автоматического выполнения заданного кода при наступлении событий изменения данных (INSERT, UPDATE, DELETE). Они обеспечивают сложную бизнес-логику, аудит и расширенную валидацию.

Отличия по типам:

BEFORE — срабатывает до физической записи изменений в таблицу. Позволяет модифицировать или проверить вводимые значения прямо «на лету» до того, как они попадут на диск.

AFTER — срабатывает после успешной записи данных в таблицу. Используется для каскадных обновлений в других таблицах, логирования действий или отправки уведомлений, когда факт вставки/изменения уже подтвержден.

INSTEAD OF — срабатывает вместо запланированного действия. Чаще всего применяется для обновления сложных нередактируемых представлений (Views).

Практическое занятие 10. Что такое индекс в СУБД, каков принцип его работы и в каких случаях создание индекса может замедлить работу с базой данных?

Что это и принцип работы: Индекс — это отдельная служебная структура (обычно в виде B-Дерева), привязанная к столбцу таблицы, которая работает как алфавитный указатель в книге. Вместо полного последовательного сканирования всей таблицы (Sequential Scan), СУБД мгновенно находит адрес нужной строки по индексу (Index Scan), резко ускоряя операции чтения данных (SELECT).

Когда индекс замедляет работу: Индексы замедляют операции модификации данных — INSERT, UPDATE, DELETE. При каждом изменении, добавлении или удалении строки СУБД вынуждена одновременно перестраивать и обновлять структуру самого индекса на диске. Также избыток индексов тратит оперативную память и дисковое пространство.

Практическое занятие 11. Что означает аббревиатура ACID в контексте транзакций СУБД? Опишите кратко каждое свойство.

Атомарность: Транзакция выполняется по принципу «всё или ничего». Если хоть одна команда внутри транзакции завершилась сбоем, вся транзакция отменяется (ROLLBACK), возвращая базу в исходное состояние.

Согласованность: Транзакция переводит базу данных из одного корректного (согласованного) состояния в другое. Все правила, ограничения (CHECK, внешние ключи) должны строго соблюдаться по завершении транзакции.

Изолированность: Параллельно выполняющиеся транзакции не должны влиять друг на друга. Результаты транзакции невидимы для других пользователей до тех пор, пока она не будет зафиксирована (COMMIT).

Долговечность: Если транзакция завершена успешно (COMMIT), внесенные изменения записываются на диск гарантированно. Никакие последующие сбои питания или крушение сервера не могут уничтожить эти данные.

Практическое занятие 12. Каковы различия между ролевой моделью доступа и индивидуальными правами пользователей? С помощью каких SQL-операторов они настраиваются?

При индивидуальных правах привилегии (чтение, запись) выдаются вручную конкретному пользователю. Это неудобно при расширении штата, так как администрировать каждого сотрудника приходится отдельно.

При ролевой модели (RBAC) права привязываются к созданной абстрактной роли (например, роль Operator, роль Manager). Затем пользователи просто назначаются членами этой роли и мгновенно наследуют все ее права. Это стандарт автоматизации администрирования.

SQL-операторы: GRANT (выдать права/роль), REVOKE (отозвать права), DENY (явный запрет доступа, имеет наивысший приоритет в некоторых СУБД, например MS SQL Server).

Практическое занятие 13. Чем отличаются друг от друга полный (Full), дифференциальный (Differential) бэкапы и бэкап журнала транзакций (Transaction Log / WAL)?

Полный бэкап (Full): Копирует абсолютно все данные и структуру базы данных. Является основой любой стратегии, но занимает много места и требует много времени на создание.

Дифференциальный бэкап (Differential): Копирует только те данные, которые изменились с момента создания последнего полного бэкапа. Занимает меньше места и создается быстрее, чем полный.

Бэкап журнала транзакций (Transaction Log / WAL в PostgreSQL): Копирует только историю выполненных сервером команд (логов) с момента последнего бэкапа любого типа. Занимает минимум места, создается мгновенно и позволяет восстановить базу с точностью до секунды.

Практическое занятие 14. Что такое восстановление базы данных на определенный момент времени (Point-in-Time Recovery) и какие файлы для этого необходимы?

Это процесс восстановления базы данных до точного хронологического момента в прошлом (например, на состояние вчера в 14:05:12). Применяется, если пользователь случайно удалил критическую таблицу или произошел сбой.

Необходимые файлы:

1. Последний полный бэкап, сделанный до указанного момента времени.

2. Все последовательные файлы журналов транзакций (Transaction Logs / WAL), накопленные в период времени от этого полного бэкапа до целевой секунды сбоя.

Практическое занятие 15. Что представляет собой угроза типа «SQL-инъекция» (SQL Injection) и какие методы защиты от нее применяются на уровне СУБД и кода?

Атака, при которой злоумышленник внедряет вредоносный SQL-код в текстовые поля ввода веб-приложения. Из-за отсутствия проверки этот код напрямую передается СУБД и выполняется, позволяя хакеру украсть данные, обойти авторизацию или удалить базу.

Методы защиты:

Использование параметризованных (подготовленных) запросов (Prepared Statements): СУБД жестко разделяет тело запроса и данные. Вводимый пользователем текст воспринимается строго как строка, а не как исполняемая команда.

Использование хранимых процедур: Скрытие прямой структуры таблиц и работа через защищенный программный интерфейс СУБД.

Принцип наименьших привилегий: Ограничение прав учетной записи, из-под которой веб-приложение подключается к СУБД (например, запрет веб-пользователю выполнять команду DROP TABLE).

Ответы полностью готовы к интеграции в ФОС. Что м

Тестовые задания для текущего контроля

Задания открытого типа

Раздел 1: Архитектура и установка MS SQL Server.

Вопрос 1. Каково назначение системной базы данных tempdb?

Опишите, какие типы объектов в ней хранятся и почему для нее критически важна высокая производительность дисковой подсистемы (I/O).

Раздел 2: Проектирование и DDL.

Вопрос 2. Объясните разницу между физическим удалением связанных данных с помощью опции ON DELETE CASCADE и занулением ссылок через ON DELETE SET NULL в ограничении FOREIGN KEY. К каким изменениям в структуре дочерней таблицы это обязывает проектировщика?

Раздел 3: Манипуляция данными и выборки.

Вопрос 3. Опишите, в каком логическом порядке MS SQL Server обрабатывает операторы FROM, WHERE, GROUP BY, HAVING и SELECT. Объяснить, почему использование агрегатных функций (например, SUM, AVG) недопустимо внутри секции WHERE.

Раздел 4: Манипуляция данными и выборки.

Вопрос 4. Сформулируйте разницу в алгоритмах работы и итоговом результате между операторами UNION и UNION ALL при объединении двух выборок. Как это влияет на использование оперативной памяти сервером?

Раздел 5: Программирование на T-SQL

Вопрос 5. Поясните, в чем заключается разница между скалярной пользовательской функцией (Scalar UDF) и встроенной табличной функцией (Inline

Table-Valued UDF) с точки зрения производительности и формирования плана выполнения запроса оптимизатором.

Раздел 6: Программирование на T-SQL.

Вопрос 6. Какую роль в DML-триггерах играют виртуальные таблицы inserted и deleted? Опишите, как будет выглядеть содержимое обеих таблиц в момент фиксации транзакции оператором UPDATE.

Раздел 7: Индексация и оптимизация.

Вопрос 7. В чем заключается физическое различие хранения данных на диске при наличии в таблице кластеризованного индекса (Clustered Index) и при его отсутствии (структура Куча — Heap)? Сколько кластеризованных индексов может иметь одна таблица и почему?

Раздел 8: Индексация и оптимизация.

Вопрос 8. Объяснить феномен Key Lookup (или RID Lookup) в плане выполнения запроса. С помощью какой опции при создании некластеризованного индекса можно избавиться от этой дорогостоящей операции?

Раздел 9: Транзакции и блокировки.

Вопрос 9. Опишите феномен «неповторяющегося чтения» (Non-repeatable Read) в условиях конкурентного доступа к данным. Какой минимальный уровень изолированности транзакций (Isolation Level) гарантирует защиту от этой проблемы?

Раздел 10: Администрирование и безопасность.

Вопрос 10. Объясните техническое различие между понятиями Login (Имя входа) и User (Пользователь базы данных) в экосистеме безопасности MS SQL Server. Как они связаны между собой?

Задания закрытого типа

1. Какая системная база данных MS SQL Server используется Агентом (SQL Server Agent) для хранения расписаний, заданий (Jobs) и планов обслуживания?

А) master

Б) model

В) msdb

Г) tempdb

2. Какое ограничение целостности в MS SQL Server автоматически создает уникальный индекс, но в отличие от первичного ключа позволяет хранить одно значение NULL в столбце?

А) CHECK

Б) UNIQUE

В) FOREIGN KEY

Г) DEFAULT

3. При использовании какого оператора соединения (JOIN) в итоговую выборку гарантированно попадут все строки из левой таблицы, даже если для них нет соответствий в правой таблице?

А) INNER JOIN

Б) LEFT OUTER JOIN

В) CROSS JOIN

Г) FULL OUTER JOIN

4. С помощью какой команды T-SQL можно принудительно сгенерировать пользовательскую ошибку и передать управление в блок CATCH?

А) RETURN

Б) THROW

В) ROLLBACK

Г) BREAK

5. Внутри какого объекта программирования в MS SQL Server категорически запрещено использовать операторы управления транзакциями (COMMIT и ROLLBACK)?

А) В хранимой процедуре (Stored Procedure)

Б) В пользовательской функции (UDF)

В) В триггере типа AFTER

Г) В триггере типа INSTEAD OF

6. Какая операция в графическом плане выполнения запроса свидетельствует о полном последовательном сканировании страниц данных таблицы, не имеющей кластеризованного индекса?

А) Index Seek

Б) Clustered Index Scan

В) Table Scan

Г) Key Lookup

7. Какую структуру данных по умолчанию использует MS SQL Server для построения стандартных индексов?

А) Хэш-таблица (Hash table)

Б) Сбалансированное Б-Дерево (B-Tree)

В) Граф (Graph)

Г) Бинарное дерево поиска (BST)

8. Какой уровень изолированности транзакций полностью защищает от феномена «фантомного чтения» (Phantom Read) путем наложения блокировок на диапазон значений (Range locks)?

А) READ UNCOMMITTED

Б) READ COMMITTED

В) REPEATABLE READ

Г) SERIALIZABLE

9. Какая модель восстановления (Recovery Model) базы данных MS SQL Server отключает возможность резервного копирования журнала транзакций (Log Backup), автоматически усекая его при каждом фиксации транзакции?

А) FULL

Б) SIMPLE

В) BULK-LOGGED

Г) ARCHIVE

10. С помощью какого оператора можно предоставить пользователю права на выполнение конкретной хранимой процедуры?

А) GRANT EXECUTE

Б) ALLOW EXECUTE

В) REVOKE EXECUTE

Г) DENY EXECUTE

Перечень практических заданий к промежуточной аттестации

Задание 1. DDL-скрипт для создания таблицы Employees (Сотрудники) и таблицы Departments (Отделы). Обеспечьте ссылочную целостность (FOREIGN KEY). Добавьте ограничение CHECK, которое гарантирует, что заработная плата сотрудника не может быть меньше 20 000 рублей, а дата приема на работу не может быть позже текущей системной даты.

Задание 2. Манипуляция данными и выборки.
Напишите SQL-запрос для вывода списка всех клиентов, которые не сделали ни одного заказа за последний календарный год. Запрос должен использовать соединение таблиц Clients и Orders, группировку, а также фильтрацию с помощью HAVING.

Задание 3. Программирование на T-SQL.
Напишите хранимую процедуру sp_UpdateProductPrice, принимающую на вход идентификатор категории товара и процент изменения цены. Процедура должна обновлять цены товаров в таблице Products в рамках явной транзакции (BEGIN TRANSACTION). Реализуйте блок TRY...CATCH: в случае любой системной ошибки

транзакция должна быть полностью откатана, а описание ошибки возвращено пользователю через THROW.

Задание 4. Программирование на T-SQL.
Разработайте триггер trg_Audit_Salary типа AFTER UPDATE на таблицу Staff. При изменении оклада сотрудника триггер должен автоматически записывать факт изменения в таблицу аудита Salary_History, фиксируя: ID сотрудника, старый оклад, новый оклад, имя системного пользователя, выполнившего операцию, и точное время.

Задание 5. Администрирование и безопасность.
Напишите комплексный T-SQL скрипт администрирования:

1. Создайте роль базы данных ReportViewer.
2. Предоставьте этой роли права на чтение (SELECT) для всех таблиц схемы dbo.
3. Создайте пользователя базы данных AnalystUser, привязанного к существующему SQL-логину, и включите его в состав созданной роли ReportViewer.

Пример задания для дифференцированного зачета

Теоретическая часть

Вопрос 1. Каковы основные архитектурные отличия реляционной модели данных от NoSQL решений (например, документоориентированных)?

Вопрос 2. Какие этапы включает в себя процесс нормализации данных до 3-й нормальной формы (3NF)?

Вопрос 3. В чем разница между системными базами данных master и model в MS SQL Server?

Вопрос 4. Опишите синтаксис и логику работы операторов CROSS APPLY и OUTER APPLY в T-SQL.

Вопрос 5. Для чего используются обобщенные табличные выражения (CTE — Common Table Expressions) и в чем их преимущество перед подзапросами?

Вопрос 6. Что такое динамический SQL (Dynamic SQL)? Каковы риски его использования (SQL-инъекции) и как их минимизировать?

Вопрос 7. Объясните принцип работы стратегий соединения таблиц Nested Loops, Merge Join и Hash Match при оптимизации планов выполнения.

Вопрос 8. Как работает механизм безопасности на уровне отдельных строк таблицы (Row-Level Security — RLS)?

Вопрос 9. Назовите типы резервного копирования в MS SQL Server (Full, Differential, Log) и опишите типичный недельный план бэкапов для базы данных в модели восстановления FULL.

Вопрос 10. Что такое взаимоблокировка (Deadlock) и какие методы существуют для её предотвращения и разрешения силами СУБД?

Практическая часть

Задание 1. Напишите запрос с использованием оконных функций (ROW_NUMBER() или DENSE_RANK()), который выводит топ-3 самых дорогих товаров в каждой отдельной категории из таблицы Products.

Задание 2. Создайте много операторную табличную функцию (Multi-statement TVF), которая принимает диапазон дат и возвращает агрегированную таблицу с финансовыми итогами продаж за каждый день этого периода.

Задание 3. Имеется медленный запрос, выполняющий поиск по текстовому полю Email и числовому полю StatusID. Напишите T-SQL команду создания некластеризованного индекса, который будет покрывать этот запрос и содержать поле FIO в секции INCLUDE.

Задание 4. Напишите скрипт, изменяющий уровень изолированности текущей сессии на SNAPSHOT. Объясните, какие требования к базе данных (настройки базы данных) должны быть выполнены для корректной работы этого режима.

Задание 5. Напишите скрипт для генерации полной резервной копии базы данных EnterpriseDB на диск D:\Backups\ с использованием опций сжатия (COMPRESSION) и проверки контрольной суммы (CHECKSUM).

Пример задания для экзамена

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Вопрос 1. Архитектура транзакций: требования ACID. Механизм работы журнала транзакций (.ldf) и стратегия упреждающей записи WAL в MS SQL Server.

Вопрос 2. Индексация данных: фундаментальные отличия между кластеризованным и некластеризованным индексами. Влияние избыточной индексации на операции INSERT, UPDATE, DELETE.

Вопрос 3. Модели восстановления баз данных (SIMPLE, FULL, BULK-LOGGED). Ограничения и возможности резервного копирования, накладываемые каждой моделью.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Напишите встроенную табличную функцию (Inline TVF) с именем «fn_GetEmployeeOrders», которая принимает на вход ID сотрудника и возвращает таблицу со столбцами: Имя сотрудника, Количество оформленных им заказов и Общая сумма этих заказов. Данные берутся из таблиц Employees и Orders. Использовать группировку.

Задание 2. Напишите T-SQL скрипт, который анализирует структуру существующей таблицы «Invoices». Скрипт должен создать некластеризованный индекс по столбцу «InvoiceDate», включающий в себя столбцы «TotalSum» и «CustomerID» (покрывающий индекс) для оптимизации поиска отчетов за отчетный период.

4 Сведения о фонде оценочных средств и его согласование

Фонд оценочных средств для аттестации по учебному модулю ПМ.02 «Администрирование баз данных» представляет собой компонент основной образовательной программы среднего профессионального образования по 09.02.12 «Техническая эксплуатация и сопровождение информационных систем»».

Фонд оценочных средств рассмотрен и одобрен на заседании методической комиссии «Информационных систем и программирования»

Протокол № 9 от «20» мая 2026 г.

Председатель методической комиссии _____/Т.Н.Богатырева /