

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Самолетостроения и эксплуатации авиационной техники
(104)»

УТВЕРЖДЕНА:
на заседании кафедры
Протокол №9 от 18 мая 2026 г.

Рабочая программа дисциплины

«БАЗЫ ДАННЫХ»

Направление: 25.03.01 Техническая эксплуатация летательных аппаратов и двигателей

Техническое обслуживание летательных аппаратов и авиационных двигателей

Квалификация: Бакалавр

Форма обучения: заочная

Документ подписан простой электронной подписью Составитель программы: Мироненко Владимир Витальевич Дата подписания: 17.06.2026
--

Документ подписан простой электронной подписью Утвердил: Подрез Никодим Владимирович Дата подписания: 19.06.2026
--

Документ подписан простой электронной подписью Согласовал: Бобарика Игорь Олегович Дата подписания: 18.06.2026
--

Год набора – 2026

Иркутск, 2026 г.

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

1.1 Дисциплина «Базы данных» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ОПК ОС-2 Способность применять современные информационные технологии при решении задач профессиональной деятельности	ОПК ОС-2.3

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ОПК ОС-2.3	Использует основные системные и прикладные программные средства для представления информации в требуемом формате	Знать классификацию и архитектуру современных СУБД (реляционные, документные, колоночные, графовые, временные ряды), основы реляционной алгебры и нормализации, синтаксис основных команд SQL (DDL, DML, SELECT с агрегацией, JOIN, оконные функции, CTE), принципы построения ER-диаграмм, механизмы транзакций и уровни изоляции, способы индексирования и оптимизации запросов, а также базовые методы защиты от SQL-инъекций и разграничения прав доступа — всё это для последующего использования прикладных системных и программных средств (MySQL, Workbench, DBeaver) для формирования структурированных выборок данных в авиационной предметной области. Уметь устанавливать и настраивать СУБД (на примере MySQL), создавать и модифицировать таблицы с помощью DDL, заполнять их данными через DML, строить сложные аналитические запросы с фильтрацией, группировкой, соединениями и оконными функциями, проектировать базы данных по ER-модели для учёта объектов авиационной техники (самолёты, двигатели, агрегаты, полётные

		параметры), разрабатывать представления, хранимые процедуры и триггеры для автоматизации контроля ресурсов, управлять транзакциями и пользовательскими ролями, а также подготавливать данные для экспорта в требуемых форматах (CSV, JSON) с использованием графических IDE и терминала. Владеть практическими навыками работы в среде MySQL и клиентских приложениях (Workbench, DBeaver), методиками нормализации и денормализации схем, приёмами оптимизации запросов с использованием EXPLAIN, способами обработки больших объёмов временных рядов (партиционирование), техникой создания рекурсивных CTE для иерархических структур, а также основами интеграции базы данных с внешними скриптами (например, через Python) для выгрузки данных в форматы, пригодные для дальнейшего анализа и построения цифровых двойников.
--	--	---

2 Место дисциплины в структуре ООП

Изучение дисциплины «Базы данных» базируется на результатах освоения следующих дисциплин/практик: «Информационные технологии», «Математика», «Введение в профессиональную деятельность»

Дисциплина является предшествующей для дисциплин/практик: «Основы теории надежности», «Безопасность полетов и сохранение летной годности», «Техническая диагностика», «Основы теории технической эксплуатации летательных аппаратов»

3 Объем дисциплины

Объем дисциплины составляет – 3 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Учебный год № 2
Общая трудоемкость дисциплины	108	108
Аудиторные занятия, в том числе:	14	14
лекции	6	6
лабораторные работы	8	8
практические/семинарские занятия	0	0

Самостоятельная работа (в т.ч. курсовое проектирование)	85	85
Трудоемкость промежуточной аттестации	9	9
Вид промежуточной аттестации (итогового контроля по дисциплине)	Экзамен	Экзамен

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Учебный год № 2

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Лекция 1 и Лекция 2	1	1	1, 2	2			1, 2, 3	14	Тест
2	Лекция 3 и Лекция 4	2	1	3, 4	2			1, 2, 3	14	Тест
3	Лекция 5	3	1	5	1			1, 2, 3	14	Тест
4	Лекция 6	4	1	6	1			1, 2, 3	14	Тест
5	Лекция 7	5	1	7	1			1, 2, 3	14	Тест
6	Лекция 8	6	1	8	1			1, 2, 3	15	Тест
	Промежуточная аттестация								9	Экзамен
	Всего		6		8				94	

4.2 Краткое содержание разделов и тем занятий

Учебный год № 2

№	Тема	Краткое содержание
1	Лекция 1 и Лекция 2	Понятие базы данных и СУБД. Эволюция: от файловых систем к современным хранилищам. Классификация по модели данных: Иерархические, сетевые (исторический аспект). Реляционные (классика): таблицы, строки, столбцы, первичный/внешний ключ. Постреляционные (NoSQL): документные (MongoDB), ключ-значение (Redis), колоночные (ClickHouse), графовые (Neo4j), временные ряды (TimescaleDB, InfluxDB). Обзор «зоопарка» СУБД (упоминание всех перечисленных вами систем с группировкой): Тяжелый корпоративный сегмент: Oracle, IBM DB2 (LUW/z/OS), Microsoft SQL Server. Open-source аналоги: PostgreSQL, MySQL/MariaDB, SQLite. Аналитические (OLAP)

		и MPP-системы: Greenplum, Vertica, Snowflake, BigQuery, ClickHouse. NoSQL и специализированные: Cassandra, Couchbase, Neo4j, Redis, Elasticsearch. Китайские/азиатские СУБД: Dameng, GaussDB, OceanBase, TiDB. Облачные и гибридные: Amazon Redshift, Athena, YugabyteDB, CockroachDB. Выбор СУБД в инженерии: почему в авиации чаще используют PostgreSQL, Oracle и специализированные временные БД (для параметров полета). Реляционная алгебра (основы). Нормальные формы (1НФ – 3НФ, НФБК). Когда денормализация нужна (применительно к большим данным). Физическая и логическая независимость данных. Типы данных (числовые, строковые, JSON, геопространственные — актуально для карт полетов). Индексы как основа производительности: B-tree, Hash, Bitmap. Чем отличаются индексы в MySQL (InnoDB) и PostgreSQL. Связи между таблицами (1:1, 1:M, M:M) на примере справочников авиатехники (типы ВС, модели двигателей, детали). Демонстрация ER-диаграммы для базы данных учета наработки агрегатов.
2	Лекция 3 и Лекция 4	Группы команд SQL: DDL (CREATE/ALTER/DROP), DML (INSERT/UPDATE/DELETE/SELECT), DCL (GRANT/REVOKE). Синтаксис SELECT и фильтрация (WHERE, LIKE, IN). Агрегация (COUNT, SUM, AVG, MIN, MAX). Группировка (GROUP BY) и HAVING. Пример из авиации: поиск всех полетов борта, где средний расход топлива превышает норму; подсчет количества отказов по типам деталей за месяц. Особенности типов данных NULL в расчетах надежности. «Фишки» популярных СУБД: как работают оконные функции в PostgreSQL vs MySQL (в MySQL их добавили позже). Сравнение диалектов. Типы соединений (INNER, LEFT, RIGHT, FULL, CROSS). Разница между ними на примере объединения таблиц «Парк ВС» и «Планы ТО». Подзапросы (скалярные, табличные) и их оптимизация. Оконные функции (ROW_NUMBER, RANK, LAG, LEAD) для анализа временной последовательности (например, изменение параметров двигателя от полета к полету). CTE (WITH) и рекурсивные CTE. Рекурсия для построения иерархии (структура самолета: фюзеляж -> крыло -> механизация -> узлы крепления). Специфика работы с временными рядами: В чем отличие запросов к ClickHouse (колоночная) от MySQL. Использование

		секционирования по времени (Partition by date) в PostgreSQL. Обзор NoSQL-запросов (MongoDB Query API, Neo4j Cypher) в сравнении с SQL — понимание, что это не одно и то же.
3	Лекция 5	Проектирование структуры БД для учета экземпляров ВС, серийных номеров, комплектации двигателями и агрегатами. Справочники (авиационные библиотеки): ATA-коды (системы ВС по стандарту ATA 100). Справочник деталей (P/N — Part Number) с альтернативными заменами. Связь с конструкторской документацией: хранение ссылок на PDF/чертежи (использование типа BLOB или ссылок в файловом хранилище). Реализация иерархии (рекурсивный CTE) для дерева разборки самолета (от планера до болта). Представления (VIEW) для упрощения: «Вид полной информации по борту» (JOIN 5-6 таблиц) и «Вид текущего состояния ресурсов».
4	Лекция 6	Сбор телеметрии (бортовые параметры: температура, вибрация, обороты). Использование временных баз данных: когда нужны TimescaleDB или InfluxDB (специализированные под временные ряды), а когда — обычный PostgreSQL с партиционированием. Расчет наработки ресурсов: написание запроса, который автоматически суммирует налет часов для каждого агрегата. Триггеры для автоматического пересчета статуса детали при достижении предельного ресурса (например, при вставке нового полета триггер обновляет признак «Требуется замена»). Анализ надежности: группировка отказов по модели ВС/двигателю с использованием агрегатных функций для построения отчетов.
5	Лекция 7	ACID в авиационной БД (критичность: нельзя потерять запись о замене агрегата). Уровни изоляции: что такое потерянное обновление и фантомы в контексте одновременной работы нескольких инженеров (смена статусов заявок). Блокировки и Deadlock: как избежать взаимных блокировок в системе планирования ТО. Права доступа: разграничение ролей (Инженер-конструктор видит всё, Техник видит только вверенный борт, Администратор управляет справочниками). Оптимизация тяжелых отчетов: Анализ плана выполнения (EXPLAIN). Сравнение производительности JOIN vs подзапрос в PostgreSQL. Денормализация (заранее вычисленные таблицы) для агрегированных отчетов по надежности. Вопросы масштабирования: шардирование (как в YugabyteDB/CockroachDB) для федеральных баз

		данных всех ВС РФ.
6	Лекция 8	Что такое база знаний в отличие от базы данных (правила, продукционная модель, онтологии). Графовые БД (Neo4j) как основа для базы знаний. Пример базы знаний: правила назначения периодичности ТО в зависимости от условий эксплуатации (пыльный аэродром, морской климат). Хранимые процедуры и функции, реализующие сложную бизнес-логику (автоматическое формирование графика ТО). Интеграция с Python/R для анализа больших данных (сквозь JDBC/ODBC). Выгрузка данных для обучения моделей машинного обучения (прогнозирование отказов). Цифровой двойник самолета: связка CAD/CAE-данных с эксплуатационной БД. Хранение истории модификаций конструкции. Краткий обзор SQL-инъекций и защиты (пункт 11.4) — почему нельзя доверять пользовательскому вводу. Итоговый синопсис: как выбрать СУБД для авиационной задачи (от MySQL для небольшого склада до Hadoop/Hive для исторических данных по 50-летней эксплуатации).

4.3 Перечень лабораторных работ

Учебный год № 2

№	Наименование лабораторной работы	Кол-во академических часов
1	Установка и первичная настройка СУБД MySQL. Знакомство со средой разработчика	1
2	Проектирование реляционной схемы базы данных	1
3	Наполнение базы данных и выполнение простых аналитических запросов	1
4	Построение сложных аналитических запросов с соединениями, оконными функциями и рекурсивными CTE	1
5	Разработка представлений и хранимых процедур для комплексного учёта воздушных судов	1
6	Автоматизация контроля ресурсов и мониторинга состояния с использованием триггеров	1
7	Управление транзакциями, разграничение прав доступа и оптимизация запросов	1
8	Интеграция баз данных с внешними скриптами, защита от SQL-инъекций и работа с большими данными	1

4.4 Перечень практических занятий

Практических занятий не предусмотрено

4.5 Самостоятельная работа

Учебный год № 2

№	Вид СРС	Кол-во академических часов
1	Оформление отчетов по лабораторным и практическим работам	37
2	Подготовка к практическим занятиям (лабораторным работам)	30
3	Тестирование по разделам дисциплин	18

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Дискуссия, Проект, Кейс-технология

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по лабораторным работам:

Лабораторные работы дисциплины построены по принципу «от простого к сложному» и охватывают полный жизненный цикл разработки авиационной базы данных — от установки СУБД до интеграции с внешними скриптами. Каждая работа является логическим продолжением предыдущей: спроектированная на втором занятии схема наполняется данными, обрастает запросами, представлениями, триггерами и процедурами, а затем оптимизируется и защищается от угроз. Студентам рекомендуется вести единый отчёт, фиксируя все этапы, листинги SQL-кода, ER-диаграммы, скриншоты результатов выполнения запросов и планы оптимизации (EXPLAIN), что позволит системно представить итоговый проект.

При выполнении работ особое внимание следует уделять корректности именования объектов (стиль SQL), выбору типов данных (особенно для серийных номеров, дат и JSON-полей) и использованию индексов на часто фильтруемых столбцах. Все запросы должны быть проверены на тестовых наборах данных, сгенерированных самостоятельно или с помощью онлайн-сервисов; при этом необходимо осознанно подходить к нормализации (не превышая ЗНФ без веских причин) и денормализации для ускорения отчётных выборок. В каждой работе предусмотрены сравнительные задания (например, сравнение JOIN и подзапросов, триггеров и ручного обновления), по результатам которых требуется сформулировать выводы о производительности и удобстве использования той или иной конструкции.

Защита каждой лабораторной работы проходит в форме короткого собеседования, где студент демонстрирует работающую базу данных, поясняет принятые проектные решения и отвечает на контрольные вопросы по теории (нормализация, транзакции, уровни изоляции, виды индексов, синтаксис оконных функций). Отчёт должен содержать титульный лист, постановку задачи, описание ER-модели, скрипты создания и наполнения таблиц, тексты всех разработанных запросов и процедур, а также скриншоты результатов с пояснениями. Работы выполняются индивидуально; использование готовых решений без собственного осмысления не допускается, так как каждая последующая работа опирается на результаты предыдущей.

5.1.2 Методические указания для обучающихся по самостоятельной работе:

Самостоятельная работа по дисциплине включает два обязательных компонента: тестирование по каждой из восьми лекций и оформление отчётов по лабораторным работам. Тестирование проводится в письменной или электронной форме после изучения соответствующего лекционного материала; для успешной сдачи необходимо освоить не только конспект, но и дополнительную литературу (например, документацию MySQL, статьи по оптимизации), а также уметь сопоставлять теоретические положения с практическими примерами из авиационной области. Рекомендуется заранее прорабатывать тестовые вопросы по каждому разделу, используя материалы лекций, лабораторные задания и самостоятельно найденные источники, чтобы закрепить понимание классификации СУБД, синтаксиса SQL, механизмов транзакций и методов защиты.

Оформление отчётов по лабораторным работам должно выполняться аккуратно и строго по структуре: цель работы, задание, ход выполнения (с листингами кода и скриншотами), анализ результатов и выводы. Каждый отчёт сдаётся не позднее недели после выполнения соответствующей лабораторной работы; при этом допускается использование единого сквозного отчёта, который пополняется по мере прохождения всех восьми работ. Важно включать в отчёт не только успешные запросы, но и примеры ошибочных ситуаций (например, нарушение внешнего ключа, deadlock) с описанием способов их устранения — это повышает практическую ценность отчёта и демонстрирует глубокое понимание материала.

Трудоёмкость самостоятельной работы распределена равномерно: на каждую лекцию выделено 2 часа на подготовку к тесту и 2–4 часа на оформление отчёта (в зависимости от сложности работы). Рекомендуется планировать самостоятельные занятия сразу после аудиторных, чтобы свежие знания сразу закреплялись практическим оформлением. Для подготовки к тестам полезно использовать краткие конспекты, выполненные в ходе лекций, а также решать типовые задачи на написание SQL-запросов. Все отчёты и результаты тестирования являются основанием для допуска к экзамену, поэтому их выполнение обязательно для всех студентов.

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 учебный год 2 | Тест

Описание процедуры.

Промежуточное тестирование по разделам дисциплины проводится в электронной информационно-образовательной среде, поддерживающей удалённый доступ через веб-интерфейс, оптимизированный для использования на персональных компьютерах, планшетах и смартфонах. Студенты получают индивидуальные учётные записи и пароли, после авторизации им открывается доступ к тестовым заданиям по каждой из восьми лекционных тем в строго отведённый временной интервал. Тест состоит из вопросов закрытого типа (с выбором одного или нескольких правильных ответов), открытых вопросов (краткий текстовый ответ) и задач на написание SQL-кода, которые проверяются

автоматически с использованием встроенных средств сравнения эталонных результатов и синтаксического анализа. Система фиксирует дату, время начала и завершения тестирования, количество попыток (предусмотрено не более двух повторных попыток для повышения результата), а по окончании формирует детализированный протокол с указанием правильных и ошибочных ответов, набранных баллов и процента выполнения. Отчёты о результатах тестирования выгружаются в формате PDF и сохраняются в личном кабинете преподавателя для последующего анализа успеваемости и корректировки учебного процесса.

Критерии оценивания.

Оценка каждого теста осуществляется по 100-балльной шкале, где каждый вопрос и задача имеют заранее определённую «весовую» стоимость в зависимости от сложности (например, простые вопросы оцениваются в 2–3 балла, задачи на написание запросов — до 10 баллов). Минимальный порог для зачёта по отдельному тесту установлен на уровне 60 % от максимального балла; при первой неудачной попытке студенту предоставляется право повторного прохождения в течение трёх календарных дней с заменой вопросов на эквивалентные по тематике, однако итоговый балл засчитывается как среднее арифметическое между результатами всех попыток. Для допуска к экзамену необходимо набрать не менее 60 % по каждому из восьми тестов (зачётная оценка «удовлетворительно»); студенты, набравшие 80–89 %, получают «хорошо», а 90–100 % — «отлично». Автоматическая система проверки генерирует подробный отчёт, в котором отображается не только итоговый балл, но и распределение ошибок по тематическим разделам, что позволяет студенту увидеть свои слабые места и целенаправленно подготовиться к пересдаче или экзамену. При возникновении технических сбоев во время тестирования предусмотрена процедура апелляции с повторным прохождением в присутствии преподавателя с использованием резервного варианта.

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ОПК ОС-2.3	Оценка сформированности компетенции проводится по результатам выполнения восьми лабораторных работ (каждая оценивается по шкале «зачёт/незачёт» с учётом корректности схемы, правильности SQL-запросов, эффективности оптимизации и оформления отчёта), а также на промежуточной аттестации (экзамен), где студент решает практическую задачу по проектированию БД и написанию запросов к авиационной тематике, демонстрируя знание теории и умение применять изученные	Для проверки знаний и умений используются: тестирование по теоретическим разделам (30 вопросов с выбором ответа), практическое задание на компьютере с установленной MySQL (создание базы по описанию,

	средства; итоговый балл складывается из среднего балла за лабораторные работы и результата экзаменационного задания.	написание 5–7 запросов разной сложности, включая JOIN, группировку, оконные функции), устное собеседование по вопросам оптимизации и администрирования, а также защита отчётов по всем лабораторным работам с демонстрацией кода и объяснением принятых проектных решений — все эти методы интегрированы в единый экзаменационный билет, позволяющий комплексно оценить достижение индикатора ОПК ОС-2.4.
--	---	--

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Учебный год 2, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.1.1 Описание процедуры

Промежуточная аттестация по дисциплине «Базы данных и знаний» проводится в форме экзамена, включающего две обязательные части: электронное тестирование и выполнение практического задания. Первая часть (тестирование) осуществляется удалённо с использованием веб-платформы, доступной через браузер на персональных устройствах студентов, включая смартфоны. Тестовый набор содержит 30 вопросов закрытого и открытого типов, охватывающих все темы курса (классификация СУБД, реляционная модель, синтаксис DDL/DML/DCL, сложные запросы, транзакции, оптимизация, базы знаний). На выполнение теста отводится 45 минут; система автоматически фиксирует ответы, формирует протокол с разбивкой по темам и предоставляет возможность однократного повторного прохождения (с другим вариантом) для улучшения результата в

течение трёх дней после первой попытки. Результаты тестирования конвертируются в баллы (максимум 40 баллов) и служат допуском ко второй части экзамена.

Вторая часть экзамена представляет собой практическое задание, выполняемое в присутствии преподавателя (очно или дистанционно с демонстрацией экрана). Студент получает индивидуальное описание предметной области (например, учёт полётов и отказов для конкретного типа ВС) и должен на работающем сервере MySQL создать недостающие таблицы, написать 5–7 запросов различной сложности (JOIN, агрегация, оконные функции, подзапросы) и пояснить принятые проектные решения. На выполнение практической части отводится 60 минут; разрешается пользоваться справочной документацией MySQL и собственными конспектами, но запрещено использование готовых скриптов из интернета. Преподаватель оценивает корректность синтаксиса, логическую правильность результатов, эффективность запросов (использование индексов, отсутствие избыточных операций) и способность аргументировать выбор конструкции. По завершении обеих частей выставляется итоговая оценка, при этом успешное прохождение тестирования (не менее 60 % правильных ответов) является обязательным условием для допуска к практической части.

Пример задания:

Пример экзаменационного билета (вариант 3):

Теоретический вопрос: «Сравните реляционные и графовые базы данных на примере задачи хранения иерархической структуры самолёта (агрегаты, узлы, детали). В каких случаях применение графовой СУБД (Neo4j) предпочтительнее реляционной, а в каких – уступает? Приведите обоснование с точки зрения производительности и сложности запросов.»

Практическое задание: Дана база данных авиационного предприятия, содержащая таблицы: Aircraft (id, model, reg_number, manufacture_date), Flight (id, aircraft_id, flight_date, flight_hours, fuel_consumption), Component (id, name, aircraft_id – ссылка на экземпляр ВС, resource_limit_hours, current_hours), Failure (id, component_id, flight_id, failure_date, description, severity). Требуется:

Написать запрос, выводящий для каждого борта (reg_number) суммарный налёт часов и средний расход топлива за последние 6 месяцев.

Написать запрос с оконной функцией, показывающий для каждого компонента дату последнего отказа и наработку в часах на момент этого отказа (используя LAG или LEAD).

Создать представление (VIEW), которое отображает полную информацию по компонентам с истёкшим ресурсом (current_hours >= resource_limit_hours), включая модель ВС и количество отказов за весь срок службы.

Написать хранимую процедуру, которая при вставке нового полёта автоматически обновляет current_hours для всех компонентов данного ВС и, если ресурс превышен, заносит запись в таблицу Alerts с уровнем критичности.

Второй абзац примера: Для выполнения задания студенту необходимо предварительно ознакомиться со структурой существующих таблиц (скрипт создания выдаётся вместе с билетом), затем набрать и отладить запросы в среде MySQL Workbench или DBeaver. После завершения кодирования студент должен устно пояснить, почему выбраны именно те или иные типы JOIN, какие индексы целесообразно создать для ускорения запросов, и

как можно оптимизировать процедуру при большом количестве компонентов (например, использовать пакетное обновление). В ответе также требуется указать возможные альтернативные решения (например, использование триггера вместо процедуры) и сравнить их с предложенным вариантом. Проверка правильности запросов осуществляется на заранее подготовленном тестовом наборе данных, результаты которых студент должен предъявить преподавателю в виде скриншотов или вывода консоли.

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Отлично : Студент демонстрирует безупречное знание теоретического материала, свободно классифицирует СУБД, обосновывает выбор модели данных для авиационных задач, безошибочно отвечает на все вопросы теста (не менее 90 % правильных ответов). Практическое задание выполнено полностью: все запросы синтаксически и логически верны, эффективно используют индексы и оптимальные конструкции (оконные функции, СТЕ, рекурсию), хранимая процедура работает корректно и	Хорошо : Студент уверенно владеет основными теоретическими разделами, допускает незначительные неточности в классификации или трактовке нормальных форм, но в целом отвечает на вопросы теста правильно (75–89 %). Практическое задание выполнено в полном объеме, однако допущены мелкие синтаксические ошибки, исправленные после замечания, или один из запросов не оптимален по производительности (например, отсутствует нужный индекс или использован подзапрос вместо JOIN). Пояснения даны, но не всегда детализированы; студент не предложил	Удовлетворительно : Студент показывает знание базовых понятий (реляционная модель, основные команды SQL, простые запросы), тест пройден с результатом 60–74 %. Практическое задание выполнено частично: два из трёх запросов работают, но содержат существенные логические ошибки, приводящие к неверной выборке; процедура создана, но не обновляет данные корректно при всех условиях. Студент испытывает затруднения при пояснении выбора конструкций, не может предложить оптимизацию, но после наводящих вопросов исправляет явные ошибки. Допускается использование справочных материалов без глубокого понимания.	Неудовлетворительно : Студент не усвоил базовые разделы дисциплины: тест пройден с результатом ниже 60 %, теоретические вопросы вызывают серьёзные затруднения (не может объяснить назначение первичного ключа или JOIN). Практическое задание не выполнено или содержит грубые ошибки (запросы не работают, синтаксис нарушен, процедура отсутствует или завершается с ошибкой). Студент не способен пояснить свои действия, не понимает логики построения запросов и не может исправить ошибки даже с помощью преподавателя. В этом случае экзамен считается несданным, и требуется повторная аттестация.

обрабатывает краевые случаи, студент даёт развёрнутые пояснения по каждому решению, предлагает варианты улучшения производительнос- ти и аргументированно сравнивает альтернативные подходы.	альтернатив, но способен ответить на уточняющие вопросы преподавателя.		
---	--	--	--

7 Основная учебная литература

1. Фуфаев Э. В. База данных : учебное пособие для студентов учреждений среднего профессионального образования / Э. В. Фуфаев, Д. Э. Фуфаев, 2009. - 320.

8 Дополнительная учебная литература и справочная

1. Библиотечно-библиографическая классификация. Средние таблицы: База данных ББК [Электронный ресурс] . Вып. 1 - 6. Дополнительный выпуск, 2014. - 1 электрон. опт. диск (CD-ROM)
2. Универсальная десятичная классификация [Электронный ресурс] : База данных, 2013. - 1 электрон. опт. диск (CD-ROM)

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Лицензионное программное обеспечение Системное программное обеспечение
2. Лицензионное программное обеспечение Пакет прикладных офисных программ
3. Лицензионное программное обеспечение Интернет-браузер

4. Свободно распространяемое программное обеспечение mysql-8.4.9
5. Свободно распространяемое программное обеспечение mysql-workbench-community-8.0.47
6. Свободно распространяемое программное обеспечение dbeaver-ce-26.1.0

12 Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения лекционных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.
2. Учебная аудитория для проведения лабораторных/практических (семинарских) занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.