

**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 И ЗНАНИЙ»

Специальность: 24.05.07 Самолето-и вертолетостроение

Самолетостроение

Квалификация: Инженер

Форма обучения: очная

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: 18.06.2026
--

Документ подписан простой электронной подписью Согласовал: Распопина Вера Борисовна Дата подписания: 17.06.2026

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 и знаний» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПК-1 Способность и готовность участвовать в разработке проектов авиационной техники различного целевого назначения	ПК-1.8

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПК-1.8	Разрабатывает базы данных объектов, относящихся к профессиональной сфере деятельности для решения инженерных задач	<p>Знать методологии проектирования баз данных (инфологическое и даталогическое моделирование), принципы нормализации и денормализации применительно к инженерным задачам авиационной отрасли, классификацию и архитектурные особенности современных СУБД (реляционные, документные, колоночные, графовые, временных рядов), полный синтаксис языка SQL (DDL, DML, DCL, TCL) с акцентом на создание сложных запросов (JOIN, подзапросы, оконные функции, рекурсивные CTE), механизмы поддержки целостности данных (первичные и внешние ключи, ограничения, триггеры), способы индексирования и анализа планов выполнения, а также базовые принципы управления транзакциями, уровнями изоляции и разграничения прав доступа для обеспечения надёжного хранения и обработки данных об объектах профессиональной сферы (воздушные суда, двигатели, агрегаты, полётная информация).</p> <p>Уметь самостоятельно анализировать предметную область, выделять сущности и связи, строить ER-диаграммы и преобразовывать их в реляционные схемы с соблюдением нормальных форм (до 3НФ), разрабатывать физическую</p>

		<p>структуру базы данных в среде MySQL (создание таблиц, индексов, ограничений), наполнять её тестовыми данными с помощью команд DML, формулировать и выполнять аналитические запросы с фильтрацией, группировкой, агрегацией, соединениями и оконными функциями для получения отчётов по надёжности, ресурсам и техническому состоянию, проектировать представления (VIEW), хранимые процедуры и пользовательские функции для автоматизации типовых инженерных расчётов, а также управлять транзакциями и ролевой моделью доступа, обеспечивая целостность и безопасность данных при коллективной работе.</p> <p>Владеть практическими навыками работы в среде MySQL и графических клиентах (Workbench, DBeaver), методами нормализации и осознанной денормализации схем для оптимизации производительности, приёмами использования различных типов индексов (B-tree, Hash, полнотекстовые) и анализа планов выполнения запросов (EXPLAIN), техникой построения рекурсивных иерархических структур (CTE) для моделирования дерева комплектации ВС, способами эффективного хранения и обработки временных рядов (партиционирование, секционирование), инструментами импорта/экспорта данных в распространённые форматы (CSV, JSON), а также основами интеграции базы данных с внешними скриптами (на примере Python) для выгрузки результатов расчётов в системы цифрового двойника или поддержки принятия решений.</p>
--	--	--

2 Место дисциплины в структуре ООП

Изучение дисциплины «Базы данных и знаний» базируется на результатах освоения следующих дисциплин/практик: «Информационные технологии», «Математика»

Дисциплина является предшествующей для дисциплин/практик: «Интеллектуальные системы в самолетостроении», «Технология производства самолета (вертолета)», «Гидропривод», «Динамика полета самолета», «Разработка прикладных модулей САПР», «Силовая установка»

3 Объем дисциплины

Объем дисциплины составляет – 3 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 6
Общая трудоемкость дисциплины	108	108
Аудиторные занятия, в том числе:	48	48
лекции	16	16
лабораторные работы	0	0
практические/семинарские занятия	32	32
Самостоятельная работа (в т.ч. курсовое проектирование)	60	60
Трудоемкость промежуточной аттестации	0	0
Вид промежуточной аттестации (итогового контроля по дисциплине)	Зачет	Зачет

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 6

№ 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	1	2			1	4	1, 2	7	Тест
2	Лекция 2	2	2			2	4	1, 2	7	Тест
3	Лекция 3	3	2			3	4	1, 2	7	Тест
4	Лекция 4	4	2			4	4	1, 2	7	Тест
5	Лекция 5	5	2			5	4	1, 2	7	Тест
6	Лекция 6	6	2			6	4	1, 2	7	Тест
7	Лекция 7	7	2			7	4	1, 2	9	Тест
8	Лекция 8	8	2			8	4	1, 2	9	Тест
	Промежуточная аттестация									Зачет
	Всего		16				32		60	

4.2 Краткое содержание разделов и тем занятий

Семестр № 6

№	Тема	Краткое содержание
1	Лекция 1	<p>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).</p>
2	Лекция 2	<p>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.</p>
3	Лекция 3	<p>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.</p>

		«Фишки» популярных СУБД: как работают оконные функции в PostgreSQL vs MySQL (в MySQL их добавили позже). Сравнение диалектов.
4	Лекция 4	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.
5	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».
6	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.
7	Лекция 7	ACID в авиационной БД (критичность: нельзя

		<p>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.</p>
8	Лекция 8	<p>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).</p>

4.3 Перечень лабораторных работ

Лабораторных работ не предусмотрено

4.4 Перечень практических занятий

Семестр № 6

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Установка и первичная настройка СУБД	4

	MySQL. Знакомство со средой разработчика	
2	Проектирование реляционной схемы базы данных	4
3	Наполнение базы данных и выполнение простых аналитических запросов	4
4	Построение сложных аналитических запросов с соединениями, оконными функциями и рекурсивными CTE	4
5	Разработка представлений и хранимых процедур для комплексного учёта воздушных судов	4
6	Автоматизация контроля ресурсов и мониторинга состояния с использованием триггеров	4
7	Управление транзакциями, разграничение прав доступа и оптимизация запросов	4
8	Интеграция баз данных с внешними скриптами, защита от SQL-инъекций и работа с большими данными	4

4.5 Самостоятельная работа

Семестр № 6

№	Вид СРС	Кол-во академических часов
1	Оформление отчетов по лабораторным и практическим работам	28
2	Тестирование по разделам дисциплин	32

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 практическим занятиям

Практические занятия 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 практической 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 семестр 6 | 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
ПК-1.8	Оценка сформированности компетенции ПК-1.8 производится на	Для проверки достижения

	<p>основе совокупности результатов выполнения восьми лабораторных работ (каждая оценивается по шкале «зачёт/незачёт» с учётом корректности схемы, правильности и эффективности SQL-кода, полноты отчёта) и итогового экзамена, включающего теоретическое тестирование и практическое задание по разработке базы данных для заданного фрагмента авиационной предметной области. Для положительной аттестации необходимо набрать не менее 60 % от максимального балла в каждой части; при этом учитываются не только синтаксическая правильность запросов, но и их производительность (наличие оптимальных индексов, отсутствие избыточных операций), а также способность студента аргументировать выбранные проектные и оптимизационные решения. Итоговая оценка выставляется как средневзвешенный балл за лабораторные работы (40 %), тестирование (30 %) и практическое задание с собеседованием (30 %)</p>	<p>компетенции ПК-1.8 используются: электронное тестирование по всем теоретическим разделам дисциплины (классификация СУБД, нормализация, синтаксис DDL/DML/DCL, транзакции, оптимизация) с автоматической генерацией протокола результатов; практическое задание на компьютере с установленным MySQL, где студент за 60 минут создаёт недостающие таблицы, пишет 5–7 запросов различной сложности (включая JOIN, агрегацию, оконные функции, подзапросы) и разрабатывает хранимую процедуру или триггер; устное собеседование по защите лабораторных работ, в ходе которого студент демонстрирует работающие скрипты, поясняет схему БД и отвечает на</p>
--	--	--

		<p>уточняющие вопросы по оптимизации и выбору конструкций. Все эти методы интегрированы в единую экзаменационную процедуру, что позволяет комплексно оценить как теоретическую подготовку, так и практические навыки разработки БД для инженерных задач профессиональной сферы.</p>
--	--	---

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 6, Типовые оценочные средства для проведения зачета 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

Зачтено	Не зачтено
----------------	-------------------

<p>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 обрабатывает краевые случаи, студент даёт развёрнутые пояснения по каждому решению, предлагает варианты улучшения производительности и аргументированно сравнивает альтернативные подходы.</p>	<p>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.</p>
---	--

7 Основная учебная литература

1. Дорофеев А. С. Базы данных : учебное пособие для специальности "Вычислительные машины, комплексы, системы и сети" / А. С. Дорофеев, 2008. - 99.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-2524.pdf>

2. Сосинская С. С. Базы данных : учебное пособие / С. С. Сосинская, 2002. - 115.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-2280.pdf>

3. Работа с базами данных [Электронный ресурс] : методические указания по выполнению лабораторных работ по курсу "Базы данных" для студентов специальностей 130100 и 130300 / Иркутский гос. технический ун-т, 2003. - 14.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-6546.pdf>

4. Кузин А. В. Базы данных : учебное пособие по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / А. В. Кузин, С. В. Левонисова, 2010. - 311.

5. Дударева. Информационное обеспечение, базы данных [Электронный ресурс] : учебное пособие: в 2 ч. Ч. 2, 2010. - 64.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-4364.pdf>

8 Дополнительная учебная литература и справочная

1. Ульман Дж. Базы данных на Паскале / Дж. Ульман, 1990. - 368.

2. История инженерного дела [Электронный ресурс] : библиографическая и полнотекстовая базы данных / Б-ка Иркут. гос. техн. ун-та, при финансовом содействии ин-та "Открытое о-во" (Фонд Сороса), 2001. - 1 электрон. опт. диск (CD-ROM)

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-6142.pdf>

3. Базы данных [Электронный ресурс] : лабораторный практикум: методические указания по лабораторным работам для специальностей 160200 "Авиастроение", специальностей: 160201 "Самолето- и вертолетостроение" 160901 "Техническая эксплуатация летательных аппаратов и двигателей" / Иркут. гос. техн. ун-т, 2007. - 21.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-5606.pdf>

4. Голицына О. Л. Базы данных : учебное пособие для вузов по направлению 230700 "Прикладная информатика" / О. Л. Голицына, Н. В. Максимов, И. И. Попов, 2012. - 399.

5. Гришин Г. Г. Базы данных на предприятиях автомобильного транспорта : учебное пособие для студентов направления 190600.62 - Эксплуатация транспортно-технологических машин и комплексов, профиль "Автомобили и автомобильное хозяйство" / Г. Г. Гришин, 2013. - 26.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-6177.pdf>

6. Молокова С. В. Базы данных и сетевые технологии : учебное пособие / С. В. Молокова, В. Б. Распопина, 2013. - 139.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-2651.pdf>

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. Свободно распространяемое программное обеспечение mysql-8.4.9
2. Лицензионное программное обеспечение Системное программное обеспечение
3. Лицензионное программное обеспечение Пакет прикладных офисных программ
4. Лицензионное программное обеспечение Интернет-браузер
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.