

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ**
Основы проектирования баз данных

Код модуля
1156466(1)

Модуль
Основания информационных технологий II часть

Екатеринбург

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Егоров Павел Владимирович	без ученой степени, без ученого звания	Старший преподаватель	департамент математики, механики и компьютерных наук
2	Созыкин Андрей Владимирович	кандидат технических наук, без ученого звания	Доцент	информационных технологий и систем управления

Согласовано:

Управление образовательных программ

Ю.Д. Маева

Авторы:

- Егоров Павел Владимирович, Старший преподаватель, департамент математики, механики и компьютерных наук
- Созыкин Андрей Владимирович, Доцент, информационных технологий и систем управления

1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Основы проектирования баз данных

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Контрольная работа	2
		Домашняя работа	3

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Основы проектирования баз данных

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ОПК-4 -Способен решать задачи профессиональной деятельности с использованием информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Д-1 - Демонстрировать развитие компетенций в области ИТ З-1 - Формулировать представления о роли современных информационно-коммуникационных технологий для решения задач профессиональной деятельности	Домашняя работа № 2 Домашняя работа № 3 Домашняя работа №1 Зачет Контрольная работа № 2 Контрольная работа №1 Лабораторные занятия
ОПК-5 -Способен использовать существующие программные продукты и информационные	Д-1 - Демонстрировать развитие компетенций в области ИТ З-1 - Сравнивать возможности различных современных программных средств для	Домашняя работа № 2 Домашняя работа № 3 Домашняя работа №1 Зачет Контрольная работа № 2 Контрольная работа №1

базы данных для решения задач профессиональной деятельности	сбора, передачи, обработки и накопления информации З-2 - Сделать обзор возможностей использования информационных баз в профессиональной деятельности П-1 - Предлагать пути решения задач по профилю деятельности, используя адекватное программное обеспечение П-2 - Иметь опыт решения задач профессиональной деятельности с использованием современных информационных баз данных У-1 - Осуществлять выбор адекватного программного обеспечения при решении задач по профилю деятельности У-2 - Осуществлять поиск и выбор необходимых информационных баз данных для решения профессиональных задач	Лабораторные занятия
ПК-2 -Способен разрабатывать, реализовывать и управлять процессами жизненного цикла программных продуктов	З-1 - Характеризовать и классифицировать различные процессы жизненного цикла программных продуктов У-1 - Выбирать и определять оптимальные методы для реализации жизненного цикла программных продуктов	Домашняя работа № 2 Домашняя работа № 3 Домашняя работа №1 Зачет Контрольная работа № 2 Контрольная работа №1 Лабораторные занятия

3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

3.1. Процедуры текущей и промежуточной аттестации по дисциплине

1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – не предусмотрено		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

Весовой коэффициент значимости результатов текущей аттестации по лекциям – не предусмотрено		
Промежуточная аттестация по лекциям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – не предусмотрено		
2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям– не предусмотрено		
Промежуточная аттестация по практическим/семинарским занятиям–нет		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям– не предусмотрено		
3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий –1.00		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	4	20
<i>домашняя работа</i>	7	20
<i>домашняя работа</i>	15	20
<i>контрольная работа</i>	10	20
<i>контрольная работа</i>	17	20
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям –0.50		
Промежуточная аттестация по лабораторным занятиям –зачет		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – 0.50		
4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий –не предусмотрено		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям –не предусмотрено		
Промежуточная аттестация по онлайн-занятиям –нет		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– **не предусмотрено**

Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – **не предусмотрено**

4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)			
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания	
		Традиционная характеристика уровня	Качественная характеристика уровня

1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

5.1.1. Лабораторные занятия

Примерный перечень тем

1. Основы SQL
2. Группировка и объединение
3. Запросы. Расширенные возможности
4. Таблицы и Связи
5. Представления, Триггеры
6. Транзакции
7. Блокировки
8. Оптимизация запросов

LMS-платформа

1. <https://ulearn.me/course/db/>

5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

Базовый

5.2.1. Контрольная работа №1

Примерный перечень тем

1. Основы SQL. Группировка и объединение. Запросы. Расширенные возможности
- Примерные задания

Вы попробовали открыть дверь. Ничего не вышло. Но в вас все еще есть частичка надежды. Вы решили удалить все привилегии открытия дверей у всех пользователей кроме себя.

Таблица user_privileges

Столбец	Тип	Модификаторы	Описание
user_id	int	not null	Идентификатор пользователя
privilege	varchar(20)	not null	Название привилегии
description	varchar(255)		Описание
expires	timestamp	not null	Время, до которого действует

Задание

Удалите привилегии `OpeningDoor` и `OpeningSecureDoor`, у которых идентификатор пользователя не равен `1618033`. При удалении верните идентификаторы пользователей, у которых были удалены соответствующие привилегии.

```
1
2
3
4
```

Как называется и сколько содержит мест самый вместительный самолёт?

Формат вывода: Название самолета | К-во мест.

Все нужные данные можно найти в представлении `aircrafts` и в таблице `seats`.

```
1
2
3
4
```

Получите номера 5 ближайших не задержанных рейсов, доступные для регистрации и где есть непришедшие на рейс пассажиры.

Можно считать, что пассажир не пришел на рейс, если у него нет посадочного талона.

```
1
2
3
4
```

LMS-платформа

1. <https://ulearn.me/course/db/>

5.2.2. Контрольная работа № 2

Примерный перечень тем

1. Таблицы и Связи. Представления, Триггеры. Блокировки

Примерные задания

Оказалось, что иногда нужно удалять статьи. Чтобы избежать потери данных, вы решили просто добавить статус «Удалён» у существующего поля `published_status`.

Описание	Название	Тип	Обязательность	Комментарий
Опубликовано	<code>published_status</code>	Число	+	По умолчанию — Не опубликовано.

Варианты значений:

- -1 — удалёно
- 0 — не опубликовано
- 1 — опубликовано

Задание

1. [Освежи знания по документации.](#)
2. [Посмотри синтаксис команды ALTER TABLE.](#)
3. Измени тип колонки `published` так, чтобы в нём можно было хранить 3 разных статуса. Не потеряйте данные в текущей таблице.
4. Добавь ограничение с названием `published_values` на возможные значения.
5. Переименуй колонку.
6. Получи все статьи, отсортированные по ID. Обрати внимание на данные в колонке со статусом.

Необходимо реализовать функциональность для уведомления приложения об изменении данных в таблице `flights`, для сброса кэша. Функция `notify_on_data_change` уже есть в БД, нужно только настроить ее вызов при любых модификациях таблицы `flights`.

```
1 -- Метод уведомления приложения
2 CREATE OR REPLACE FUNCTION notify_on_data_change() RETURNS trigger
3 AS $BODY$
4 BEGIN
5     PERFORM pg_notify('data_changed', TG_TABLE_NAME);
6     RAISE NOTICE 'Сообщение об изменении данных успешно отправлено';
7     RETURN NEW;
8 END
9 $BODY$
10 LANGUAGE 'plpgsql';
```

```
1
2
3
4
```

Посмотри в документации раздел "Предложение блокировки" как использовать блокировки на уровне строк у команды SELECT.

Задание

Получи информацию по бронированию `D8094E`. Назначь получаемым строкам **минимально-достаточный** режим блокировки, чтобы их не могли изменить из другой транзакции. Строки, которые нельзя заблокировать нужно пропускать.

Формат вывода: номер бронирования | номер билета | стоимость перелета

```
1
2
3
4
```

LMS-платформа

1. <https://ulearn.me/course/db/>

5.2.3. Домашняя работа №1

Примерный перечень тем

1. Основы SQL

2. Группировка и объединение

Примерные задания

Задание

Реализуй возможность добавления или обновления посадочного талона.

1. Скачай проект
2. Нужно реализовать метод `CreateOrUpdate` в классе `BoardingPassProducer`
 - Если у пассажира нет посадочного талона, его нужно создать и вернуть
 - Если у пассажира есть талон, то нужно обновить у него место и вернуть
3. Для проверки можешь запускать тесты в файле `BoardingPassProducerTests.cs`
 - Для подключения к БД используются настройки по-умолчанию и пароль postgres

Для извлечения и сохранения данных из БД используйте объект контекста `AirCompanyDataContext`. Инициализировать его самостоятельно не нужно, он инжектится через конструктор в поле `_dataContext`.

```
1 // Вставьте сюда финальное содержимое файла BoardingPassProducer.cs
2
3
4
```

Необходимо написать метод для формирования отчета по завершенным перелетам, который содержит **к-во мест в самолете, к-во пассажиров, сумму цен билетов этого рейса**. В отчет должны попадать перелеты у которых **дата посадки попадает в заданный временной интервал**, и у которых **к-во пассажиров на борту было больше чем параметр `passengerCountThreshold`**. Отсортировать результаты по убыванию к-ва пассажиров.

1. Скачай проект
2. Нужно реализовать метод `Build` в классе `FlightsReportBuilder`
3. Для проверки можешь запускать тесты в файле `FlightsReportBuilderTests.cs`
 - Для подключения к БД используются настройки по-умолчанию и пароль postgres

Для извлечения и сохранения данных из БД используйте объект контекста `AirCompanyDataContext`. Инициализировать его самостоятельно не нужно, он инжектится через конструктор в поле `_dataContext`.

```
1 // Вставьте сюда финальное содержимое файла FlightsReportBuilder.cs
2
3
4
```

LMS-платформа

1. <https://ulearn.me/course/db/>

5.2.4. Домашняя работа № 2

Примерный перечень тем

1. Запросы. Расширенные возможности

2. Таблицы и Связи

Примерные задания

Для организации по контролю перелетов необходимо написать метод для формирования отчета по размещении пассажиров в салоне самолета.

На вход принимается id рейса. На выходе ожидается номер билета, номер места в самолете, класс обслуживания, ФИО и email пассажира если он есть, количество пассажиров, которое летит вместе с ним.

В отчете должен быть только один пассажир из каждого бронирования. Нужно выбирать пассажира с наименьшим номером билета.

Данные отсортировать по классу обслуживания в порядке уменьшения класса и по уменьшению количества пассажиров летящих вместе.

1. Скачай проект
2. Нужно реализовать метод `Build` в классе `FlightPassengerReportBuilder`
3. Для проверки можешь запускать тесты в файле `FlightPassengerReportBuilderTests.cs`
 - Для подключения к БД используются настройки по-умолчанию и пароль postgres

Для извлечения и сохранения данных из БД используйте объект контекста `AirCompanyDataContext`. Инициализировать его самостоятельно не нужно, он инжектится через конструктор в поле `_dataContext`.

```
1 // Вставьте сюда финальное содержимое файла FlightPassengerReportBuilder.cs
2
3
4
```

Задание

1. Скачай проект
2. Реализуй классы сущностей в `MagazineModels.cs`
3. С помощью fluent api настроить модели в `MagazineModels.cs`
4. Запустить тесты и проверить, что все тесты проходят
5. Отправить содержимое файла `MagazineModels.cs` на проверку

Скачать проект

```
1 // Вставьте сюда финальное содержимое файла MagazineModels.cs
2
3
4
```

LMS-платформа

1. <https://ulearn.me/course/db/>

5.2.5. Домашняя работа № 3

Примерный перечень тем

1. Представления, Триггеры
2. Транзакции

Примерные задания

В одном из предыдущих заданий мы делали механизм уведомления приложения об изменении данных в таблице `flights`. Теперь необходимо написать метод, который будет возвращать к-во принятых самолетов каждым аэропортом, при этом подразумевается большая нагрузка на этот метод, поэтому все данные нужно кэшировать на стороне приложения, но при обновлении данных в таблице `flights` необходимо его сбрасывать.

1. Скачай проект
2. Нужно реализовать метод `Get` и доработать метод `ListenNotificationsLoop` в классе `ArrivedAircraftCountProvider`
3. Для проверки можешь запускать тесты в файле `ArrivedAircraftCountProviderTests.cs`
 - Для подключения к БД используются настройки по-умолчанию и пароль `postgres`

Для извлечения и сохранения данных из БД используйте объект контекста `AirCompanyDataContext`. Инициализировать его самостоятельно не нужно, он инжектится через конструктор в поле `_dataContext`.

```
1 // Вставьте сюда финальное содержимое файла ArrivedAircraftCountProvider.cs
2
3
4
```

Необходимо написать метод, который будет добавлять к существующему бронированию клиента билеты на рейс. То есть тебе нужно создать новые перелеты из входных данных в таблице ``ticket_flights``.

После добавления перелетов нужно обновить у бронирования полную стоимость.

****Требования**:**

- все запросы нужно выполнить в одной транзакции;
- если во время добавления билета возникнет ошибка:
 - мы ****не**** должны потерять добавленные ранее билеты,
 - мы должны его пропустить и добавить остальные билеты;
- нужно обновить полную стоимость таким образом, чтобы не было несогласованности;

Изучи [документацию от Microsoft](<https://docs.microsoft.com/ru-ru/ef/core/saving/transactions>) про использование транзакций из кода и приступай к заданию.

1. Скачай проект
2. Нужно реализовать метод `Save` в классе `BookingTicketFlightsSaver`
3. Для проверки можешь запускать тесты в файле `BookingTicketFlightsSaverTests.cs`
 - Для подключения к БД используются настройки по-умолчанию и пароль `postgres`

Для извлечения и сохранения данных из БД используйте объект контекста `AirCompanyDataContext`. Инициализировать его самостоятельно не нужно, он инжектится через конструктор в поле `_dataContext`.

```
1 // Вставьте сюда финальное содержимое файла BookingTicketFlightsSaver.cs
2
3
4
```

LMS-платформа

1. <https://ulearn.me/course/db/>

5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

5.3.1. Зачет

Список примерных вопросов

1. С помощью какого оператора можно отфильтровать данные в запросе? Какие операторы необходимы, чтобы обновить часть данных в таблице по условию? С помощью какого оператора можно создать новую таблицу? а представление?

2. Как отсортировать данные? Как изменить порядок сортировки? Какие агрегатные функции ты знаешь? Как отфильтровать результаты группировки? Как исключить дубликаты в запросе? Какие ключи ты знаешь? Для чего они нужны? Как организовать связь многие-ко-многим?

3. Как получить данные из нескольких таблиц? (обычно отвечаю про join) Какие еще есть способы? Что такое оконная функция? Приведи примеры использования. Где использование подзапроса может приводить к очень долгому выполнению?

4. Какие есть виды нормальных форм, для чего они нужны? В какой ситуации денормализация оправдана? Что такое транзакция? Для чего она нужна? Какой принцип работы транзакции, т.е. за счет чего реализуется?

5. Какие аномалии бывают при конкурирующем выполнении транзакций? Какие уровни изоляций помогают с ними справляться? Что такое блокировки? Какие блокировки ты знаешь? Что такое покрывающий индекс?

6. Как добавить столбик в таблицу, где больше 100млн строк, новый столбец со значением по-умолчанию? Какие проблемы возникнут, если попытаться сделать это в лоб?

7. Что такое индекс? какие индексы знаешь? как они устроены? Как работает выборка по двум столбцам, на которых есть составной индекс? В каких ситуациях такой индекс будет работать, а в каких нет?

LMS-платформа – не предусмотрена

5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	профориентационная деятельность	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности	ОПК-5	З-2 У-2 П-2	Домашняя работа № 2 Домашняя работа № 3 Домашняя работа №1 Зачет Контрольная работа № 2 Контрольная работа №1 Лабораторные занятия