

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ

Руководитель образовательной программы

доц., к.т.н.

(должность, уч. степень, звание)

А.А. Фоменкова

(инициалы, фамилия)

(подпись)

«06» февраля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Проектирование баз данных»  
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности	Проектирование программных систем
Форма обучения	заочная
Год приема	2025

Лист согласования рабочей программы дисциплины

Программу составил (а)

старший преподаватель  
(должность, уч. степень, звание)

05.02.2025  
(подпись, дата)

М.В. Величко  
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«06» февраля 2025 г, протокол № 01/2025

Заведующий кафедрой № 43

д.т.н., проф.  
(уч. степень, звание)

06.02.2025  
(подпись, дата)

М.Ю. Охтилев  
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н.  
(должность, уч. степень, звание)

06.02.2025  
(подпись, дата)

А.А. Фоменкова  
(инициалы, фамилия)

## Аннотация

Дисциплина «Проектирование баз данных» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.04 «Программная инженерия» направленности «Проектирование программных систем». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-5 «Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем»

ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»

Содержание дисциплины охватывает круг вопросов, связанных с реляционными, объектными, объектно-реляционными и NoSQL СУБД, реляционной алгеброй, языком SQL, проектированием структуры баз данных, разработкой запросов к базам данных, использованием CASE-средств и систем управления базами данных для разработки баз данных, использованием современных моделей данных для разработки систем различных предметных областей.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося, контрольные работы заочников.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 7 зачетных единиц, 252 часа.

Язык обучения по дисциплине «русский».

## 1. Перечень планируемых результатов обучения по дисциплине

1.1. Целью преподавания дисциплины является получение обучающимися необходимых знаний и навыков в области организации хранения информации в базах данных в среде систем управления базами данных, обеспечения целостности данных, выполнения запросов к информации, хранимой в базах данных, получения знаний о роли баз данных в современных программных системах и современных типах систем управления базами данных.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.3.1 знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем
Общепрофессиональные компетенции	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3.1 знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-6.У.1 умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ

## 2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Объектно-ориентированное программирование»;
- «Проектирование программных систем»;
- «Алгоритмы и структуры данных».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Разработка и анализ требований»;
- «Программирование встроенных приложений»;
- «Обработка экспериментальных данных» и др.
- Дипломное проектирование.

## 3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№6	№7
1	2	3	4
<b>Общая трудоемкость дисциплины</b> , ЗЕ/ (час)	7/ 252	4/ 144	3/ 108
<b>Из них часов практической подготовки</b>			
<b>Аудиторные занятия</b> , всего час.	40	24	16
в том числе:			
лекции (Л), (час)	16	10	6
практические/семинарские занятия (ПЗ), (час)			
лабораторные работы (ЛР), (час)	24	14	10
курсовой проект (работа) (КП, КР), (час)			
экзамен, (час)	9	9	
<b>Самостоятельная работа</b> , всего (час)	203	111	92
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Зачет	Экз.	Зачет

Примечание: \*\* кандидатский экзамен

#### 4. Содержание дисциплины

##### 4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
<b>Семестр 6</b>					
Раздел 1. Введение в проектирование баз данных					
Тема 1.1. История развития и причины появления СУБД					
Тема 1.2. Модели данных					
Тема 1.3 Нормализация и денормализация баз данных					
Тема 1.4 Роль баз данных в программных системах					
	3		3		26
Раздел 2. Введение в базы данных					
Тема 2.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными					
Тема 2.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных, манипулирования данными					
Тема 2.3. Реляционная алгебра					
	3		4		38

Раздел 3. Введение в язык SQL Тема 3.1. Оператор выборки в языке SQL, агрегатные функции Тема 3.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов Тема 3.3. Использование представлений, управляющих конструкций в языке SQL Тема 3.4. Хранимые процедуры	4		7		47
Итого в семестре:	10		14		111
Семестр 7					
Раздел 3. Введение в язык SQL Тема 3.5. Триггеры, обеспечение активной целостности Тема 3.5. Управление доступом к базе данных	1				18
Раздел 4. Объектные и объектно-реляционные базы данных Тема 4.1. Выбор типа СУБД применительно к особенностям предметной области Тема 4.2. Объектно-реляционные базы данных Тема 4.3. Объектные базы данных	2		5		35
Раздел 5. NoSQL базы данных Тема 5.1. Базы данных «Ключ-значение» Тема 5.2. Документные базы данных Тема 5.3. Графовые базы данных Тема 5.4. Базы данных «Семейство столбцов»	3		5		39
Итого в семестре:	6		10		92
Итого	16	0	24	0	203

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

#### 4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Введение в проектирование баз данных Тема 1.1. История развития и причины появления СУБД Тема 1.2. Модели данных Тема 1.3 Нормализация и денормализация баз данных Тема 1.4 Роль баз данных в программных системах Тема 1.5 Установка и администрирование баз данных
2	Раздел 2. Введение в базы данных Тема 2.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными Тема 2.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания,

	удаления, модификации таблиц базы данных, манипулирования данными Тема 2.3. Реляционная алгебра
3	Раздел 3. Введение в язык SQL Тема 3.1. Оператор выборки в языке SQL, агрегатные функции Тема 3.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов Тема 3.3. Использование представлений, управляющих юнструкций в языке SQL Тема 3.4. Хранимые процедуры, триггеры, обеспечение активной целостности Тема 3.5. Управление доступом к базе данных
4	Раздел 4. Объектные и объектно-реляционные базы данных Тема 4.1. Выбор типа СУБД применительно к особенностям предметной области Тема 4.2. Объектно-реляционные базы данных Тема 4.3. Объектные базы данных
5	Раздел 5. NoSQL базы данных Тема 5.1. Базы данных «Ключ-значение» Тема 5.2. Документные базы данных Тема 5.3. Графовые базы данных Тема 5.4. Базы данных «Семейство столбцов»

#### 4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					

#### 4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 6				
1.	Разработка физической модели базы данных с учетом декларативной ссылочной целостности	3		1
2.	Создание и модификация базы данных и таблиц базы данных	2		2
3.	Заполнение таблиц и модификация данных	2		2
4.	Разработка SQL запросов: виды соединений и шаблоны	2		3
5.	Разработка SQL запросов: запросы с подзапросами	3		3
6.	Хранимые процедуры	2		3

Семестр 7				
7.	Объектно-реляционные базы данных. Проектирование и создание	3		4
8.	Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы	2		4
9.	Разработка документной базы данных	3		5
10.	Манипулирование данными в документной базе данных	2		5
Всего		24		

4.5. Курсовое проектирование/ выполнение курсовой работы

Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 6, час	Семестр 7, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	139	71	68
Подготовка к текущему контролю успеваемости (ТКУ)	16	10	6
Контрольные работы заочников (КРЗ)	14	6	8
Подготовка к промежуточной аттестации (ПА)	34	24	10
Всего:	203	111	92

#### 5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

#### 6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке ( кроме электронных экземпляров)
<a href="https://e.lanbook.com/book/176670">https://e.lanbook.com/book/176670</a> Режим доступа: для авториз. пользователей.	Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 244 с. — ISBN 978-5-8114-8412-6. - Текст : электронный.	

<a href="https://znanium.com/catalog/product/1093646">https://znanium.com/catalog/product/1093646</a> Режим доступа: по подписке	Мартишин, С. А. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с использованием Node.js, Express.js, Apache Spark и Scala : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ИНФРА-М, 2020. — 235 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cc063e18baca3.52928692. - ISBN 978-5-16-107636-1. - Текст : электронный.	
<a href="https://znanium.com/catalog/product/961470">https://znanium.com/catalog/product/961470</a> Режим доступа: по подписке.	Дадян, Э. Г. Методы, модели, средства хранения и обработки данных : учебник / Э.Г. Дадян, Ю.А. Зеленков. — М. : Вузовский учебник : ИНФРА-М, 2018. — 168 с. - ISBN 978-5-9558-0490-3. - Текст : электронный.	
<a href="https://e.lanbook.com/book/176572">https://e.lanbook.com/book/176572</a> Режим доступа: для авториз. пользователей.	Чистякова, М. А. Проектирование и эксплуатация баз данных : учебно-методическое пособие / М. А. Чистякова, И. А. Иванова, И. Д. Котилевец. — Москва : РТУ МИРЭА, 2021. — 112 с- Текст : электронный.	
<a href="https://e.lanbook.com/book/151716">https://e.lanbook.com/book/151716</a> Режим доступа: для авториз. пользователей.	Петрова, А. Н. Реализация баз данных : учебное пособие / А. Н. Петрова, В. Е. Степаненко. — Комсомольск-на-Амуре : КНАГУ, 2020. — 144 с. — ISBN 978-5-7765-1448-7. — Текст : электронный	
<a href="https://znanium.com/catalog/product/1227737">https://znanium.com/catalog/product/1227737</a> Режим доступа: по подписке.	Тарасов, С. В. СУБД для программиста: базы данных изнутри / С. В. Тарасов. - Москва : СОЛОН-Пресс, 2020. - 320 с. - ISBN 978-2-7466-7383-0. - Текст : электронный.	



<a href="https://znanium.com/catalog/product/1215513">https://znanium.com/catalog/product/1215513</a> Режим доступа: по подписке.	Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2021. — 368 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0718-4. - Текст : электронный.	
--	---	--

#### 7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
<a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a>	Электронная документация по MySQL
<a href="http://www.mysql.ru/docs/man/Reference.html">http://www.mysql.ru/docs/man/Reference.html</a>	Электронная документация по MySQL на русском языке
<a href="https://docs.mongodb.com/">https://docs.mongodb.com/</a>	Электронная документация по MongoDB
<a href="https://www.postgresql.org/docs/">https://www.postgresql.org/docs/</a>	Электронная документация по PostgreSQL

#### 8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	Не предусмотрено

8.2. Перечень информационно-справочных систем,используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

#### 9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	

2	Специализированные лаборатории кафедра 43	ул.Б.Морская 67 лит А, ауд. 23-08,23-09, 23-10
---	---	---

#### 10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Задачи; Тесты.
Зачет	Список вопросов; Тесты; Задачи.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 – Критерии оценки уровня сформированности компетенций

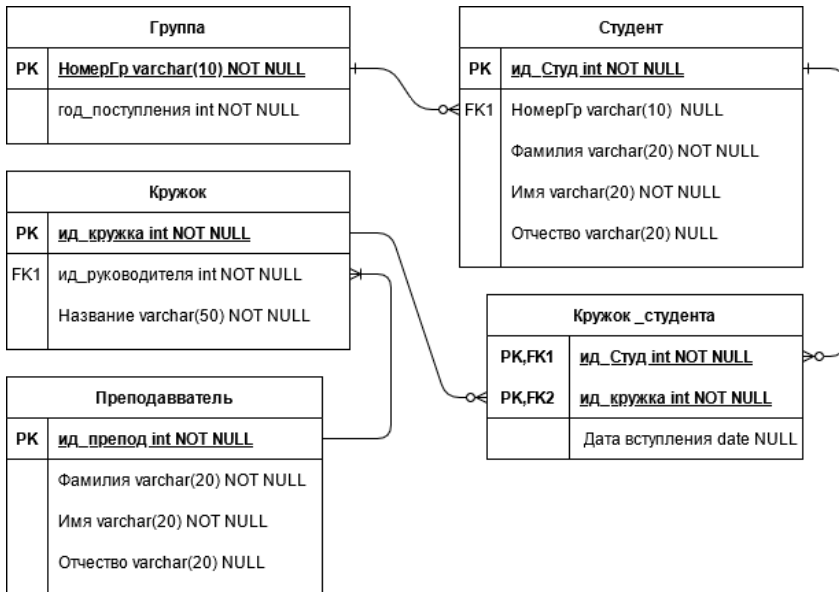
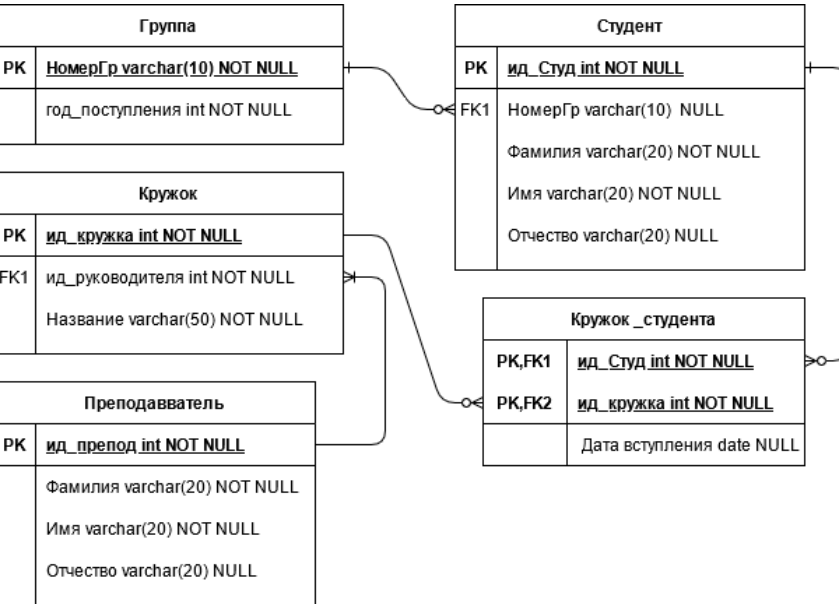
Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся глубоко и всесторонне усвоил программный материал;</li> <li>– уверенно, логично, последовательно и грамотно его излагает;</li> <li>– опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления;</li> <li>– умело обосновывает и аргументирует выдвигаемые им идеи;</li> <li>– делает выводы и обобщения;</li> <li>– свободно владеет системой специализированных понятий.</li> </ul>
«хорошо» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы;</li> <li>– не допускает существенных неточностей;</li> <li>– увязывает усвоенные знания с практической деятельностью направления;</li> <li>– аргументирует научные положения;</li> <li>– делает выводы и обобщения;</li> <li>– владеет системой специализированных понятий.</li> </ul>
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы;</li> <li>– допускает несущественные ошибки и неточности;</li> <li>– испытывает затруднения в практическом применении знаний направления;</li> <li>– слабо аргументирует научные положения;</li> <li>– затрудняется в формулировании выводов и обобщений;</li> <li>– частично владеет системой специализированных понятий.</li> </ul>
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> <li>– обучающийся не усвоил значительной части программного материала;</li> <li>– допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении;</li> <li>– испытывает трудности в практическом применении знаний;</li> <li>– не может аргументировать научные положения;</li> <li>– не формулирует выводов и обобщений.</li> </ul>

#### 10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

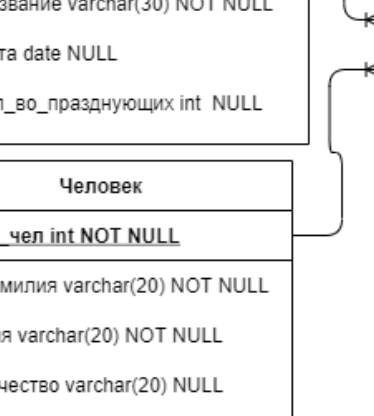
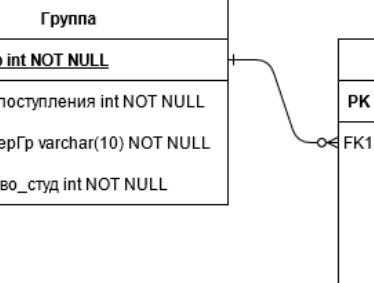

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Основные понятия БД: база данных, ИС, вычислительная система, банк данных, СУБД, словарь данных, администратор БД.	ОПК-5.3.1
2	Перечислите и охарактеризуйте функции СУБД	ОПК-5.3.1
3	Перечислите и охарактеризуйте классификации СУБД	ОПК-5.3.1
4	Дайте определение понятия «транзакция». Приведите пример транзакции. Перечислите свойства транзакции и опишите процессы журнализации и отката транзакций.	ОПК-6.3.1
5	Назовите и охарактеризуйте уровни архитектуры СУБД.	ОПК-5.3.1
6	Элементы реляционной модели БД: отношение, кортеж, атрибут, домен, значение атрибута, схема отношения, первичный ключ. Перечислите свойства отношений.	ОПК-6.3.1
7	Перечислите и охарактеризуйте виды связей между отношениями. Приведите примеры.	ОПК-6.3.1
8	Сравните понятия потенциальный, первичный и внешний ключ. Опишите процессы ограничения и каскадирования операции.	ОПК-6.3.1
9	Опишите процесс приведения БД к 1НФ	ОПК-6.3.1
10	Опишите понятия: сущность, атрибут, связь. Охарактеризуйте процесс преобразования ER-модели в реляционную БД.	ОПК-6.3.1
11	Охарактеризуйте подходы к обеспечению безопасности БД и методы управления доступом к БД.	ОПК-5.3.1
12	История причины появления СУБД. Роль СУБД в программных системах	ОПК-6.3.1
13	Объектно-ориентированные и NoSQL модели данных.	ОПК-6.3.1
14	Типы данных в языке SQL, оператор языка SQL для создания таблиц	ОПК-6.3.1
15	Операторы языка SQL для удаления и модификации таблиц	ОПК-6.3.1
16	Абстрактные операции манипулирования данными, их реализация в языке SQL	ОПК-6.3.1
17	Булевы операции над отношениями	ОПК-6.3.1
18	Операция выборки, ее свойства	ОПК-6.3.1
19	Операция проекции, ее свойства	ОПК-6.3.1
20	Операция соединения, ее свойства	ОПК-6.3.1
21	Операция деления	ОПК-6.3.1
22	Операция переименования атрибутов	ОПК-6.3.1
23	Операции эквисоединения, $\theta$ -выборки и $\theta$ -соединения	ОПК-6.3.1
24	Оператор выборки в языке SQL	ОПК-6.3.1
25	Агрегатные функции в операторе выборки языка SQL	ОПК-6.3.1
26	Объединение, пересечение, разность запросов в языке SQL	ОПК-6.3.1
27	Запросы с подзапросами в языке SQL	ОПК-6.3.1
28	Экзистенциальные запросы в языке SQL	ОПК-6.3.1
29	Виды соединений в языке SQL	ОПК-6.3.1
30	Трехзначная логика и обработка NULL-значений в языке SQL	ОПК-6.3.1
31	Представления в языке SQL	ОПК-6.3.1
32	Управляющие конструкции в языке SQL	ОПК-6.3.1
33	Хранимые процедуры в языке SQL	ОПК-6.3.1
	Задачи	
1.	Задача на написание запроса на выборку на языке SQL.	ОПК-6.У.1

	<p>Написать запрос SQL, выбирающий кружок с количеством студентов больше среднего по заданной схеме:</p>  <pre>graph LR     Group[Группа] --&gt; Student[Студент]     Group --&gt; Circle[Кружок]     Circle --&gt; Student     Circle --&gt; CircleStudent[Кружок_студента]     Teacher[Преподаватель] --&gt; Circle     Teacher --&gt; CircleStudent</pre> <p><b>Группа</b></p> <table><tr><td>PK</td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr></table> <p><b>Студент</b></p> <table><tr><td>PK</td><td>ид_Студ int NOT NULL</td></tr><tr><td>FK1</td><td>НомерГр varchar(10) NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table> <p><b>Кружок</b></p> <table><tr><td>PK</td><td>ид_кружка int NOT NULL</td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td></td><td>Название varchar(50) NOT NULL</td></tr></table> <p><b>Преподаватель</b></p> <table><tr><td>PK</td><td>ид_препод int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table> <p><b>Кружок_студента</b></p> <table><tr><td>PK,FK1</td><td>ид_Студ int NOT NULL</td></tr><tr><td>PK,FK2</td><td>ид_кружка int NOT NULL</td></tr><tr><td></td><td>Дата вступления date NULL</td></tr></table>	PK	НомерГр varchar(10) NOT NULL		год_поступления int NOT NULL	PK	ид_Студ int NOT NULL	FK1	НомерГр varchar(10) NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	PK	ид_кружка int NOT NULL	FK1	ид_руководителя int NOT NULL		Название varchar(50) NOT NULL	PK	ид_препод int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	PK,FK1	ид_Студ int NOT NULL	PK,FK2	ид_кружка int NOT NULL		Дата вступления date NULL	
PK	НомерГр varchar(10) NOT NULL																																			
	год_поступления int NOT NULL																																			
PK	ид_Студ int NOT NULL																																			
FK1	НомерГр varchar(10) NULL																																			
	Фамилия varchar(20) NOT NULL																																			
	Имя varchar(20) NOT NULL																																			
	Отчество varchar(20) NULL																																			
PK	ид_кружка int NOT NULL																																			
FK1	ид_руководителя int NOT NULL																																			
	Название varchar(50) NOT NULL																																			
PK	ид_препод int NOT NULL																																			
	Фамилия varchar(20) NOT NULL																																			
	Имя varchar(20) NOT NULL																																			
	Отчество varchar(20) NULL																																			
PK,FK1	ид_Студ int NOT NULL																																			
PK,FK2	ид_кружка int NOT NULL																																			
	Дата вступления date NULL																																			
2.	<p>Задача на написание запроса на выборку на языке SQL.</p> <p>Написать запрос SQL, выбирающий студента, который занимается в кружках всех преподавателей:</p>  <pre>graph LR     Group[Группа] --&gt; Student[Студент]     Group --&gt; Circle[Кружок]     Circle --&gt; Student     Circle --&gt; CircleStudent[Кружок_студента]     Teacher[Преподаватель] --&gt; Circle     Teacher --&gt; CircleStudent</pre> <p><b>Группа</b></p> <table><tr><td>PK</td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr></table> <p><b>Студент</b></p> <table><tr><td>PK</td><td>ид_Студ int NOT NULL</td></tr><tr><td>FK1</td><td>НомерГр varchar(10) NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table> <p><b>Кружок</b></p> <table><tr><td>PK</td><td>ид_кружка int NOT NULL</td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td></td><td>Название varchar(50) NOT NULL</td></tr></table> <p><b>Преподаватель</b></p> <table><tr><td>PK</td><td>ид_препод int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table> <p><b>Кружок_студента</b></p> <table><tr><td>PK,FK1</td><td>ид_Студ int NOT NULL</td></tr><tr><td>PK,FK2</td><td>ид_кружка int NOT NULL</td></tr><tr><td></td><td>Дата вступления date NULL</td></tr></table>	PK	НомерГр varchar(10) NOT NULL		год_поступления int NOT NULL	PK	ид_Студ int NOT NULL	FK1	НомерГр varchar(10) NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	PK	ид_кружка int NOT NULL	FK1	ид_руководителя int NOT NULL		Название varchar(50) NOT NULL	PK	ид_препод int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	PK,FK1	ид_Студ int NOT NULL	PK,FK2	ид_кружка int NOT NULL		Дата вступления date NULL	ОПК-6.У.1
PK	НомерГр varchar(10) NOT NULL																																			
	год_поступления int NOT NULL																																			
PK	ид_Студ int NOT NULL																																			
FK1	НомерГр varchar(10) NULL																																			
	Фамилия varchar(20) NOT NULL																																			
	Имя varchar(20) NOT NULL																																			
	Отчество varchar(20) NULL																																			
PK	ид_кружка int NOT NULL																																			
FK1	ид_руководителя int NOT NULL																																			
	Название varchar(50) NOT NULL																																			
PK	ид_препод int NOT NULL																																			
	Фамилия varchar(20) NOT NULL																																			
	Имя varchar(20) NOT NULL																																			
	Отчество varchar(20) NULL																																			
PK,FK1	ид_Студ int NOT NULL																																			
PK,FK2	ид_кружка int NOT NULL																																			
	Дата вступления date NULL																																			
3.	<p>Задача на написание хранимой процедуры на языке SQL.</p> <p>Для заданной схемы разработать хранимую процедуру осуществляющую каскадное удаление группы и всех её студентов и того, что с ними связано:</p>	ОПК-6.У.1																																		

	<pre> erDiagram     Группа   --o{ Студент : "участит"     Группа   --o{ Клуб : "руководит"     Студент   --o{ Клуб_студента : "участит"     Клуб   --o{ Клуб_студента : "участит"     Преподаватель   --o{ Клуб : "руководит"     Преподаватель   --o{ Клуб_студента : "участит"      Группа {         int PK идГр NOT NULL         int год_поступления NOT NULL         varchar(10) НомерГр NOT NULL     }     Студент {         int PK ид_Студ NOT NULL         int идГр NOT NULL         varchar(20) Фамилия NOT NULL         varchar(20) Имя NOT NULL         varchar(20) Отчество NOT NULL     }     Клуб {         int PK ид_клуба NOT NULL         int ид_руководителя NOT NULL         varchar(50) Название NOT NULL     }     Преподаватель {         int PK ид_препод NOT NULL         varchar(20) Фамилия NOT NULL         varchar(20) Имя NOT NULL         varchar(20) Отчество NOT NULL     }     Клуб_студента {         int PK,FK1 ид_Студ NOT NULL         int PK,FK2 ид_клуба NOT NULL         date Дата_вступления NOT NULL     } </pre>	
4	<p>Задача на написание хранимой процедуры на языке SQL.</p> <p>Для заданной схемы разработать хранимую процедуру, осуществляющую каскадное удаление праздника и того, что с ним связано:</p> <pre> erDiagram     Праздник   --o{ Празднует : "празднует"     Праздник   --o{ Подарок : "получает"     Празднует   --o{ Подаренный_подарок : "получает"     Человек   --o{ Празднует : "участит"     Человек   --o{ Подаренный_подарок : "получает"     Подарок   --o{ Подаренный_подарок : "получает"      Праздник {         int PK ид_праздник NOT NULL         varchar(30) Название NOT NULL         date Дата NOT NULL     }     Празднует {         int PK ид_чел_празд NOT NULL         int ид_праздник NOT NULL         int ид_чел NOT NULL     }     Человек {         int PK ид_чел NOT NULL         varchar(20) Фамилия NOT NULL         varchar(20) Имя NOT NULL         varchar(20) Отчество NOT NULL     }     Подарок {         int PK ид_подарка NOT NULL         varchar(30) Название NOT NULL     }     Подаренный_подарок {         int PK ид_дарения NOT NULL         int ид_чел_празд NOT NULL         int ИД_дарителя NOT NULL         int ИД_подарка NOT NULL         bool анонимно NOT NULL         date Дата NOT NULL     } </pre>	ОПК-6.У.1
5	<p>Задача на написание хранимой процедуры на языке SQL.</p> <p>Для заданной схемы разработать хранимую процедуру, осуществляющую вставку в таблицу «Празднует», с поиском и добавлением (при их отсутствии) людей и праздников по названию (все таблицы с автоинкрементными ключами):</p>	ОПК-6.У.1



	<div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>идГр int NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>кол_во_студ int NOT NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Группа		PK	<u>идГр int NOT NULL</u>		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		кол_во_студ int NOT NULL	Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	идГр int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL																							
Группа																																														
PK	<u>идГр int NOT NULL</u>																																													
	год_поступления int NOT NULL																																													
	НомерГр varchar(10) NOT NULL																																													
	кол_во_студ int NOT NULL																																													
Студент																																														
PK	<u>ид_Студ int NOT NULL</u>																																													
FK1	идГр int NOT NULL																																													
	Фамилия varchar(20) NOT NULL																																													
	Имя varchar(20) NOT NULL																																													
	Отчество varchar(20) NULL																																													
8	<div><p>Задача на написание запроса на манипулирование данными на языке SQL</p><p>Написать скрипт SQL, удаляющий кружки, в которых нет студентов из базы данных по заданной схеме:</p><div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>НомерГр varchar(10) NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr></table><table><tr><th colspan="2">Кружок</th></tr><tr><td>PK</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td></td><td>Название varchar(50) NOT NULL</td></tr></table><table><tr><th colspan="2">Преподаватель</th></tr><tr><td>PK</td><td><u>ид_препод int NOT NULL</u></td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>НомерГр varchar(10) NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table><table><tr><th colspan="2">Кружок_студента</th></tr><tr><td>PK,FK1</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>PK,FK2</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td></td><td>Дата вступления date NULL</td></tr></table></div></div>	Группа		PK	<u>НомерГр varchar(10) NOT NULL</u>		год_поступления int NOT NULL	Кружок		PK	<u>ид_кружка int NOT NULL</u>	FK1	ид_руководителя int NOT NULL		Название varchar(50) NOT NULL	Преподаватель		PK	<u>ид_препод int NOT NULL</u>		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	НомерГр varchar(10) NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Кружок_студента		PK,FK1	<u>ид_Студ int NOT NULL</u>	PK,FK2	<u>ид_кружка int NOT NULL</u>		Дата вступления date NULL	ОПК-6.У.1
Группа																																														
PK	<u>НомерГр varchar(10) NOT NULL</u>																																													
	год_поступления int NOT NULL																																													
Кружок																																														
PK	<u>ид_кружка int NOT NULL</u>																																													
FK1	ид_руководителя int NOT NULL																																													
	Название varchar(50) NOT NULL																																													
Преподаватель																																														
PK	<u>ид_препод int NOT NULL</u>																																													
	Фамилия varchar(20) NOT NULL																																													
	Имя varchar(20) NOT NULL																																													
	Отчество varchar(20) NULL																																													
Студент																																														
PK	<u>ид_Студ int NOT NULL</u>																																													
FK1	НомерГр varchar(10) NULL																																													
	Фамилия varchar(20) NOT NULL																																													
	Имя varchar(20) NOT NULL																																													
	Отчество varchar(20) NULL																																													
Кружок_студента																																														
PK,FK1	<u>ид_Студ int NOT NULL</u>																																													
PK,FK2	<u>ид_кружка int NOT NULL</u>																																													
	Дата вступления date NULL																																													
9	<div><p>Задача на написание запроса на манипулирование данными на языке SQL</p><p>Написать скрипт SQL, вставляющий в таблицу «кружки2», идентичную по структуре таблице «кружки» из базы данных по заданной схеме кружки, в которых студентов больше 5:</p><div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>НомерГр varchar(10) NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr></table><table><tr><th colspan="2">Кружок</th></tr><tr><td>PK</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td></td><td>Название varchar(50) NOT NULL</td></tr></table><table><tr><th colspan="2">Преподаватель</th></tr><tr><td>PK</td><td><u>ид_препод int NOT NULL</u></td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>НомерГр varchar(10) NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table><table><tr><th colspan="2">Кружок_студента</th></tr><tr><td>PK,FK1</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>PK,FK2</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td></td><td>Дата вступления date NULL</td></tr></table></div></div>	Группа		PK	<u>НомерГр varchar(10) NOT NULL</u>		год_поступления int NOT NULL	Кружок		PK	<u>ид_кружка int NOT NULL</u>	FK1	ид_руководителя int NOT NULL		Название varchar(50) NOT NULL	Преподаватель		PK	<u>ид_препод int NOT NULL</u>		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	НомерГр varchar(10) NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Кружок_студента		PK,FK1	<u>ид_Студ int NOT NULL</u>	PK,FK2	<u>ид_кружка int NOT NULL</u>		Дата вступления date NULL	ОПК-6.У.1
Группа																																														
PK	<u>НомерГр varchar(10) NOT NULL</u>																																													
	год_поступления int NOT NULL																																													
Кружок																																														
PK	<u>ид_кружка int NOT NULL</u>																																													
FK1	ид_руководителя int NOT NULL																																													
	Название varchar(50) NOT NULL																																													
Преподаватель																																														
PK	<u>ид_препод int NOT NULL</u>																																													
	Фамилия varchar(20) NOT NULL																																													
	Имя varchar(20) NOT NULL																																													
	Отчество varchar(20) NULL																																													
Студент																																														
PK	<u>ид_Студ int NOT NULL</u>																																													
FK1	НомерГр varchar(10) NULL																																													
	Фамилия varchar(20) NOT NULL																																													
	Имя varchar(20) NOT NULL																																													
	Отчество varchar(20) NULL																																													
Кружок_студента																																														
PK,FK1	<u>ид_Студ int NOT NULL</u>																																													
PK,FK2	<u>ид_кружка int NOT NULL</u>																																													
	Дата вступления date NULL																																													
10	<div><p>Задача на написание запроса на манипулирование данными на языке SQL</p><p>Для базы данных по заданной схеме написать скрипт SQL заполняющий корректными данными значение столбец «кол_во_празднующих»,</p></div>	ОПК-6.У.1																																												

	<p>содержащего количество людей отмечающих соответствующий праздник.</p> 	
11	<p>Задача на написание запроса на манипулирование данными на языке SQL</p> <p>Написать скрипт SQL, вставляющий в таблицу «студенты2», идентичную по структуре таблице «студенты» из базы данных по заданной схеме, студентов, которые числятся в группах, где студентов больше 20:</p> 	ОПК-6.У.1
12	<p>Задача на написание запроса на манипулирование данными на языке SQL</p> <p>Написать скрипт SQL, удаляющий праздники, которые празднует меньше 3 человек.</p> 	ОПК-6.У.1
13	<p>Задача на написание запроса на определение данных на языке SQL.</p> <p>Написать скрипт SQL, создания базы данных по заданной схеме:</p>	ОПК-6.У.1



14	<p>Задача на написание запроса на определение данных на языке SQL. Написать скрипт на SQL, изменяющий первичный ключ «НомерГр», таблицы Группа на составной первичный ключ «номерГр», «год_поступления» и все внешние ключи, затронутые изменением, соответственно.</p>	ОПК-6.У.1
15	<p>Задача на написание запроса на определение данных на языке SQL. Написать скрипт SQL, создания базы данных по заданной схеме:</p>	ОПК-6.У.1

16	<p>Задача на написание запроса на определение данных на языке SQL. Написать скрипт на SQL, изменяющий первичный ключ «ид_праздник», таблицы Праздник на составной первичный ключ «Название», «Дата» и все внешние ключи, затронутые изменением, соответственно.</p>	ОПК-6.У.1
17	<p>Задача на написание запроса на определение данных на языке SQL. Написать скрипт, создающий таблицы tab1 из 2 столбцов и tab2 из 3 столбцов. Первый столбец tab1 – строка фиксированной длины и является первичным ключом, второй – строка ограниченной длины. Первый столбец tab2 – длинное целое и является первичным ключом, второй – внешний ключ связи с tab1, третий – число с плавающей запятой одинарной точности.</p>	ОПК-6.У.1
18	<p>Задача на написание запроса на определение данных на языке SQL. Написать скрипт, создающий таблицы tab1 из 2 столбцов и tab2 из 3 столбцов. Первый столбец tab1 – строка ограниченной длины и является первичным ключом, второй – число с плавающей запятой двойной точности. Первый столбец tab2 – строка фиксированной длины и является первичным ключом, второй – внешний ключ связи с tab1, третий – число с фиксированной запятой .</p>	ОПК-6.У.1
19	<p>Задача на написание запроса на выборку на языке SQL. Написать запрос SQL, выбирающий студента, который занимается и в кружке «Олимпиадного программирования» и в кружке «Робототехники»:</p>	ОПК-6.У.1

20	<p>Задача на написание запроса на манипулирование данными на языке SQL</p> <p>Написать скрипт SQL, обновляющий в таблице «студент» отчество студентов на Иванович, если отчество пустое (пустая строка или null) и группа 2022 года поступления:</p> <div><table><tr><th colspan="2">Группа</th></tr><tr><td>ПК</td><td><u>идГр int NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>кол_во_студ int NOT NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>ПК</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Группа		ПК	<u>идГр int NOT NULL</u>		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		кол_во_студ int NOT NULL	Студент		ПК	<u>ид_Студ int NOT NULL</u>	FK1	идГр int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	ОПК-6.У.1						
Группа																														
ПК	<u>идГр int NOT NULL</u>																													
	год_поступления int NOT NULL																													
	НомерГр varchar(10) NOT NULL																													
	кол_во_студ int NOT NULL																													
Студент																														
ПК	<u>ид_Студ int NOT NULL</u>																													
FK1	идГр int NOT NULL																													
	Фамилия varchar(20) NOT NULL																													
	Имя varchar(20) NOT NULL																													
	Отчество varchar(20) NULL																													
21	<p>Задача на написание запроса на манипулирование данными на языке SQL</p> <p>Написать скрипт SQL, удаляющий праздники, которые никто не празднует.</p> <div><table><tr><th colspan="2">Праздник</th></tr><tr><td>ПК</td><td><u>ид_праздник int NOT NULL</u></td></tr><tr><td></td><td>Название varchar(30) NOT NULL</td></tr><tr><td></td><td>Дата date NULL</td></tr><tr><td></td><td>кол_во_празднующих int NULL</td></tr></table><table><tr><th colspan="2">Празднует</th></tr><tr><td>ПК</td><td><u>ид_чел_празд int NOT NULL</u></td></tr><tr><td>FK</td><td>ид_праздник int NOT NULL</td></tr><tr><td>FK</td><td>ид_чел int NOT NULL</td></tr></table><table><tr><th colspan="2">Человек</th></tr><tr><td>ПК</td><td><u>ид_чел int NOT NULL</u></td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Праздник		ПК	<u>ид_праздник int NOT NULL</u>		Название varchar(30) NOT NULL		Дата date NULL		кол_во_празднующих int NULL	Празднует		ПК	<u>ид_чел_празд int NOT NULL</u>	FK	ид_праздник int NOT NULL	FK	ид_чел int NOT NULL	Человек		ПК	<u>ид_чел int NOT NULL</u>		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	ОПК-6.У.1
Праздник																														
ПК	<u>ид_праздник int NOT NULL</u>																													
	Название varchar(30) NOT NULL																													
	Дата date NULL																													
	кол_во_празднующих int NULL																													
Празднует																														
ПК	<u>ид_чел_празд int NOT NULL</u>																													
FK	ид_праздник int NOT NULL																													
FK	ид_чел int NOT NULL																													
Человек																														
ПК	<u>ид_чел int NOT NULL</u>																													
	Фамилия varchar(20) NOT NULL																													
	Имя varchar(20) NOT NULL																													
	Отчество varchar(20) NULL																													
22	<p>Задача на написание запроса на определение данных на языке SQL.</p> <p>Написать скрипт на SQL для уже существующих таблиц, удаляющий столбец «год_поступления» таблицы «Группа» и добавляющий столбец пол в таблицу «Студент».</p> <div><table><tr><th colspan="2">Группа</th></tr><tr><td>ПК</td><td><u>НомерГр varchar(10) NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>ПК</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>НомерГр varchar(10) NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Группа		ПК	<u>НомерГр varchar(10) NOT NULL</u>		год_поступления int NOT NULL	Студент		ПК	<u>ид_Студ int NOT NULL</u>	FK1	НомерГр varchar(10) NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	ОПК-6.У.1										
Группа																														
ПК	<u>НомерГр varchar(10) NOT NULL</u>																													
	год_поступления int NOT NULL																													
Студент																														
ПК	<u>ид_Студ int NOT NULL</u>																													
FK1	НомерГр varchar(10) NULL																													
	Фамилия varchar(20) NOT NULL																													
	Имя varchar(20) NOT NULL																													
	Отчество varchar(20) NULL																													
23	<p>Задача на написание запроса на определение данных на языке SQL.</p> <p>Написать скрипт на SQL для уже существующих таблиц, удаляющий столбец «кол_во_празднующих» таблицы «Праздник» и добавляющий столбец «кол_во_праздников» в таблицу «Человек»</p>	ОПК-6.У.1																												

	<div><div><table><tr><th colspan="2">Праздник</th></tr><tr><td>PK</td><td><u>ид_праздник int NOT NULL</u></td></tr><tr><td colspan="2">Название varchar(30) NOT NULL</td></tr><tr><td colspan="2">Дата date NULL</td></tr><tr><td colspan="2">кол_во_празднующих int NULL</td></tr></table></div><div><table><tr><th colspan="2">Празднует</th></tr><tr><td>PK</td><td><u>ид_чел_празд int NOT NULL</u></td></tr><tr><td>FK</td><td>ид_праздник int NOT NULL</td></tr><tr><td>FK</td><td>ид_чел int NOT NULL</td></tr></table></div><div><table><tr><th colspan="2">Человек</th></tr><tr><td>PK</td><td><u>ид_чел int NOT NULL</u></td></tr><tr><td colspan="2">Фамилия varchar(20) NOT NULL</td></tr><tr><td colspan="2">Имя varchar(20) NOT NULL</td></tr><tr><td colspan="2">Отчество varchar(20) NULL</td></tr></table></div></div>	Праздник		PK	<u>ид_праздник int NOT NULL</u>	Название varchar(30) NOT NULL		Дата date NULL		кол_во_празднующих int NULL		Празднует		PK	<u>ид_чел_празд int NOT NULL</u>	FK	ид_праздник int NOT NULL	FK	ид_чел int NOT NULL	Человек		PK	<u>ид_чел int NOT NULL</u>	Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL																				
Праздник																																																
PK	<u>ид_праздник int NOT NULL</u>																																															
Название varchar(30) NOT NULL																																																
Дата date NULL																																																
кол_во_празднующих int NULL																																																
Празднует																																																
PK	<u>ид_чел_празд int NOT NULL</u>																																															
FK	ид_праздник int NOT NULL																																															
FK	ид_чел int NOT NULL																																															
Человек																																																
PK	<u>ид_чел int NOT NULL</u>																																															
Фамилия varchar(20) NOT NULL																																																
Имя varchar(20) NOT NULL																																																
Отчество varchar(20) NULL																																																
24	<div><p>Задача на написание хранимой процедуры на языке SQL.</p><p>Для заданной схемы разработать хранимую процедуру, осуществляющую удаление студента и удаление группы, если удаленный студент был в ней последним:</p><div><div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>идГр int NOT NULL</u></td></tr><tr><td colspan="2">год_поступления int NOT NULL</td></tr><tr><td colspan="2">НомерГр varchar(10) NOT NULL</td></tr></table></div><div><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td colspan="2">Фамилия varchar(20) NOT NULL</td></tr><tr><td colspan="2">Имя varchar(20) NOT NULL</td></tr><tr><td colspan="2">Отчество varchar(20) NULL</td></tr></table></div><div><table><tr><th colspan="2">Кружок</th></tr><tr><td>PK</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td colspan="2">Название varchar(50) NOT NULL</td></tr></table></div><div><table><tr><th colspan="2">Преподаватель</th></tr><tr><td>PK</td><td><u>ид_препод int NOT NULL</u></td></tr><tr><td colspan="2">Фамилия varchar(20) NOT NULL</td></tr><tr><td colspan="2">Имя varchar(20) NOT NULL</td></tr><tr><td colspan="2">Отчество varchar(20) NULL</td></tr></table></div><div><table><tr><th colspan="2">Кружок _студента</th></tr><tr><td>PK,FK1</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>PK,FK2</td><td><u>ид_кружка int NOT NULL</u></td></tr><tr><td colspan="2">Дата вступления date NULL</td></tr></table></div></div></div>	Группа		PK	<u>идГр int NOT NULL</u>	год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	идГр int NOT NULL	Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL		Кружок		PK	<u>ид_кружка int NOT NULL</u>	FK1	ид_руководителя int NOT NULL	Название varchar(50) NOT NULL		Преподаватель		PK	<u>ид_препод int NOT NULL</u>	Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL		Кружок _студента		PK,FK1	<u>ид_Студ int NOT NULL</u>	PK,FK2	<u>ид_кружка int NOT NULL</u>	Дата вступления date NULL		ОПК-6.У.1
Группа																																																
PK	<u>идГр int NOT NULL</u>																																															
год_поступления int NOT NULL																																																
НомерГр varchar(10) NOT NULL																																																
Студент																																																
PK	<u>ид_Студ int NOT NULL</u>																																															
FK1	идГр int NOT NULL																																															
Фамилия varchar(20) NOT NULL																																																
Имя varchar(20) NOT NULL																																																
Отчество varchar(20) NULL																																																
Кружок																																																
PK	<u>ид_кружка int NOT NULL</u>																																															
FK1	ид_руководителя int NOT NULL																																															
Название varchar(50) NOT NULL																																																
Преподаватель																																																
PK	<u>ид_препод int NOT NULL</u>																																															
Фамилия varchar(20) NOT NULL																																																
Имя varchar(20) NOT NULL																																																
Отчество varchar(20) NULL																																																
Кружок _студента																																																
PK,FK1	<u>ид_Студ int NOT NULL</u>																																															
PK,FK2	<u>ид_кружка int NOT NULL</u>																																															
Дата вступления date NULL																																																
25	<div><p>Задача на написание хранимой процедуры на языке SQL.</p><p>Для заданной схемы разработать хранимую процедуру, осуществляющую удаление из таблицы «Празднует» по «ид_чел_празд» и удаление праздника, если празднующий был единственным:</p></div>	ОПК-6.У.1																																														

26	<p>Задача на построение диаграммы «сущность-связь»</p> <p>Построить диаграммы «сущность-связь» по описанию сущностей. Нотация произвольная, но должна быть указана. Необходимо учесть правила построения концептуальной модели, не все части описания могут им соответствовать. Типы данных определяются самостоятельно. Описание сущностей в формате: имя_сущности (<b>ключевые свойства, обязательное свойство1, необязательное свойство2</b>), связей в формате: имя_связи (<b>свойство связи обязательное. свойство связи необязательное</b>): сущность1-сущность2 (мощность связи, обязательность) обязательность указывается через тире (обязательность в связи 1 сущности- обязательность 2 сущности), где О- обязательная, Н - необязательная (О-Н).</p> <p>Даны сущности: книга (<b>ид книги, ISBN, Название</b>), автор (<b>ид автора, фамилия, имя, отчество</b>), читатель (<b>ид читателя, фамилия, имя, отчество, номер паспорта</b>), читательский билет (<b>номер билета, дата выдачи</b>), издательство (<b>ид издательства, название</b>).</p> <p>Даны связи: пишет ():книга-автор (М:М,О-О),записана_на (<b>дата выдачи, дата возврата</b>): читательский билет-книга (М:М,О-Н), оформлен_на (): читатель - читательский билет(1:1,О-О), издает():издательство- книга (1:М,О-Н )</p>	ОПК-6.У.1
27	<p>Задача на построение диаграммы «сущность-связь»</p> <p>Построить диаграммы «сущность-связь» по описанию сущностей. Нотация произвольная, но должна быть указана. Необходимо учесть правила построения концептуальной модели, не все части описания могут им соответствовать. Типы данных определяются самостоятельно. Описание сущностей в формате: имя_сущности(<b>ключевые свойства, обязательное свойство1, необязательное свойство2</b>), связей в формате: имя_связи (<b>свойство связи обязательное. свойство связи необязательное</b>):сущность1-сущность2 (мощность связи, обязательность) обязательность указывается через тире (обязательность в связи 1 сущности- обязательность 2 сущности), где О- обязательная, Н - необязательная (О-Н).</p> <p>Даны сущности: кафедра (<b>ид кафедры, номер, название</b>), группа (<b>ид группы,номер</b>), дисциплина (<b>ид дисциплины, название</b>), студент (<b>ид студента, фамилия, имя ,отчество</b>).</p> <p>Даны связи: выпускает ():кафедра-группа (1:М,Н-О), учится в():группа-</p>	ОПК-6.У.1

	студент (1:M,O-O), является старостой():группа-студент (1:1,O-O),изучает( <b>количество часов</b> ): группа-дисциплина(M:M,O-H).	
28	<p>Задача на построение диаграммы «сущность-связь»</p> <p>Построить диаграммы «сущность-связь» по описанию сущностей. Нотация произвольная, но должна быть указана. Необходимо учесть правила построения концептуальной модели, не все части описания могут им соответствовать. Типы данных определяются самостоятельно. Описание сущностей в формате: имя_сущности (<b>ключевые свойства, обязательное свойство1, необязательное свойство2</b>), связей в формате: имя_связи (<b>свойство связи обязательное. свойство связи необязательное</b>): сущность1-сущность2 (мощность связи, обязательность) обязательность указывается через тире (обязательность в связи 1 сущности- обязательность 2 сущности), где О- обязательная, Н - необязательная (О-Н).</p> <p>Даны сущности: тип научной работы (<b>ид типа, название</b>), научная работа (<b>ид работы, название, УДК</b>), автор (<b>ид автора, фамилия, имя, отчество</b>),содержание работы(<b>ид содержания, аннотация, текст, ключевые слова</b>)</p> <p>Даны связи: относится к():тип научной работы- научная работа (1:M,O-H),содержит():научная работа-содержание работы1:1,O-O), является автором(<b>количество страниц автора</b>):научная работа -автор(M:M,O-O), цитирует: научная работа- научная работа(M:M,H-H).</p>	ОПК-6.У.1
29	<p>Задача на построение логической модели базы данных по концептуальной</p> <p>Преобразовать заданную ER диаграмму (концептуальную модель) в логическую. (*- обязательный атрибут, о- необязательный атрибут)</p>	ОПК-6.У.1
30	<p>Задача на построение логической модели базы данных по концептуальной</p> <p>Преобразовать заданную ER диаграмму (концептуальную модель) в логическую. (*- обязательный атрибут,о- необязательный атрибут)</p>	ОПК-6.У.1
31	<p>Задача на построение логической модели базы данных по концептуальной</p> <p>Преобразовать заданную ER диаграмму (концептуальную модель) в логическую.</p>	ОПК-6.У.1

	<div><div><div><div><div>Отряд в лагере</div><div><div>PK</div><div><u>ид_отряда int NOT NULL</u></div></div><div>Название отряда varchar(50) NOT NULL</div><div>номер смены int NOT NULL</div></div></div><div><div>состоит в</div><div><div><div></div><div></div><div></div></div></div></div><div><div>Ребенок</div><div><div>PK</div><div><u>ид_ребенка int NOT NULL</u></div></div><div>Фамилия_ребенка varchar(20) NOT NULL</div><div>Имя_ребенка varchar(20) NOT NULL</div><div>Отчество_ребенка varchar(20) NULL</div></div><div><div>имеет</div><div><div><div></div><div></div><div></div></div></div></div><div><div>вожатый</div><div><div>PK</div><div><u>ид_вожатого int NOT NULL</u></div></div><div>Фамилия_вожатого varchar(20) NOT NULL</div><div>Имя_вожатого varchar(20) NOT NULL</div><div>Отчество_вожатого varchar(20) NULL</div></div><div><div>руководит</div><div><div><div></div><div></div><div></div></div></div></div><div><div>Свидетельство о рождении</div><div><div>PK</div><div><u>ид_свидетельства int NOT NULL</u></div></div><div>номер свидетельства char(6)</div><div>дата_выдачи date NOT NULL</div></div></div></div>																									
32	<div><div><div>Задача на нормализацию базы данных</div><div>Привести предложенную базу данных к 2 нормальной форме с указанием действий по приведению к каждой нормально форме</div><div><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_студ int NOT NULL</u></td></tr><tr><td>FK</td><td>Номер группы varchar(10)</td></tr><tr><td>FK</td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>full_name(surname,name,patronym) varchar(60) NOT NULL</td></tr><tr><td></td><td>{телефон} varchar(10) NULL</td></tr></table><table><tr><th colspan="2">Группа</th></tr><tr><td>PK,FK1</td><td><u>Номер группы varchar(10) NOT NULL</u></td></tr><tr><td>PK,FK2</td><td><u>год поступления int NOT NULL</u></td></tr><tr><td></td><td>Направление varchar(10) NOT NULL</td></tr><tr><td></td><td>Группа_направлений varchar(10) NOT NULL</td></tr><tr><td></td><td>Факультет varchar(10) NOT NULL</td></tr></table></div><div><div>Зависимости кроме зависимостей от первичных ключей в формате детерминант→зависимое:</div><div>Направление→Группа_направлений, Номер группы→ факультет</div></div></div></div>	Студент		PK	<u>ид_студ int NOT NULL</u>	FK	Номер группы varchar(10)	FK	год_поступления int NOT NULL		full_name(surname,name,patronym) varchar(60) NOT NULL		{телефон} varchar(10) NULL	Группа		PK,FK1	<u>Номер группы varchar(10) NOT NULL</u>	PK,FK2	<u>год поступления int NOT NULL</u>		Направление varchar(10) NOT NULL		Группа_направлений varchar(10) NOT NULL		Факультет varchar(10) NOT NULL	ОПК-6.У.1
Студент																										
PK	<u>ид_студ int NOT NULL</u>																									
FK	Номер группы varchar(10)																									
FK	год_поступления int NOT NULL																									
	full_name(surname,name,patronym) varchar(60) NOT NULL																									
	{телефон} varchar(10) NULL																									
Группа																										
PK,FK1	<u>Номер группы varchar(10) NOT NULL</u>																									
PK,FK2	<u>год поступления int NOT NULL</u>																									
	Направление varchar(10) NOT NULL																									
	Группа_направлений varchar(10) NOT NULL																									
	Факультет varchar(10) NOT NULL																									
33	<div><div><div>Задача на нормализацию базы данных</div><div>Привести предложенную базу данных к 3 нормальной форме с указанием действий по приведению к каждой нормально форме</div><div><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_студ int NOT NULL</u></td></tr><tr><td>FK</td><td>Номер группы varchar(10)</td></tr><tr><td>FK</td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>full_name(surname,name,patronym) varchar(60) NOT NULL</td></tr><tr><td></td><td>{телефон} varchar(10) NULL</td></tr></table><table><tr><th colspan="2">Группа</th></tr><tr><td>PK,FK1</td><td><u>Номер группы varchar(10) NOT NULL</u></td></tr><tr><td>PK,FK2</td><td><u>год поступления int NOT NULL</u></td></tr><tr><td></td><td>Направление varchar(10) NOT NULL</td></tr><tr><td></td><td>Группа_направлений varchar(10) NOT NULL</td></tr><tr><td></td><td>Факультет varchar(10) NOT NULL</td></tr></table></div><div><div>Зависимости кроме зависимостей от первичных ключей в формате детерминант→зависимое:</div><div>Направление→Группа_направлений, Номер группы→ факультет</div></div></div></div>	Студент		PK	<u>ид_студ int NOT NULL</u>	FK	Номер группы varchar(10)	FK	год_поступления int NOT NULL		full_name(surname,name,patronym) varchar(60) NOT NULL		{телефон} varchar(10) NULL	Группа		PK,FK1	<u>Номер группы varchar(10) NOT NULL</u>	PK,FK2	<u>год поступления int NOT NULL</u>		Направление varchar(10) NOT NULL		Группа_направлений varchar(10) NOT NULL		Факультет varchar(10) NOT NULL	ОПК-6.У.1
Студент																										
PK	<u>ид_студ int NOT NULL</u>																									
FK	Номер группы varchar(10)																									
FK	год_поступления int NOT NULL																									
	full_name(surname,name,patronym) varchar(60) NOT NULL																									
	{телефон} varchar(10) NULL																									
Группа																										
PK,FK1	<u>Номер группы varchar(10) NOT NULL</u>																									
PK,FK2	<u>год поступления int NOT NULL</u>																									
	Направление varchar(10) NOT NULL																									
	Группа_направлений varchar(10) NOT NULL																									
	Факультет varchar(10) NOT NULL																									

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.  
Таблица 16 – Вопросы (задачи) для зачета / дифф. Зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1	Триггеры в языке SQL	ОПК-6.3.1
2	Перечислите и охарактеризуйте типы ограничений целостности БД.	ОПК-5.3.1
3	Опишите понятия инкапсуляция, наследование и полиморфизм с точки зрения теории БД.	ОПК-5.3.1
4	Дайте определение понятия целостности БД и перечислите существующие уровни изолированности транзакций.	ОПК-5.3.1
5	Выбор типа СУБД применительно к особенностям предметной области	ОПК-6.3.1
6	Объектные базы данных. (Объектная модель). Проектирование и создание	ОПК-6.3.1
7	Объектные базы данных. Манипуляции данными	ОПК-6.3.1
8	Объектно-реляционные базы данных. Проектирование и создание	ОПК-6.3.1
9	Объектно-реляционные базы данных. Пользовательские типы и операторы	ОПК-6.3.1
10	Базы данных «Ключ-значение»	ОПК-6.3.1
11	Структура и создание документных баз данных.	ОПК-
12	Манипуляции данными и поиск в документных базах данных.	ОПК-6.3.1
13	Графовые базы данных. Графовые структуры, назначение графовых бд	ОПК-6.3.1
14	Графовые базы данных : графы и операции CRUD	ОПК-6.3.1
15	Базы данных типа «Семейство столбцов». Операции CRUD и администрирование таблиц	ОПК-6.3.1
16	Базы данных типа «Семейство столбцов». работа с «большими данными»	ОПК-6.3.1



	Задачи																							
1.	<p>Для базы данных по заданной схеме написать триггеры, поддерживающие в актуальном состоянии значение столбца кол_во_студ, содержащего количество студентов группы.</p> <div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>идГр int NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>кол_во_студ int NOT NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Группа		PK	<u>идГр int NOT NULL</u>		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		кол_во_студ int NOT NULL	Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	идГр int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	ОПК-6.У.1
Группа																								
PK	<u>идГр int NOT NULL</u>																							
	год_поступления int NOT NULL																							
	НомерГр varchar(10) NOT NULL																							
	кол_во_студ int NOT NULL																							
Студент																								
PK	<u>ид_Студ int NOT NULL</u>																							
FK1	идГр int NOT NULL																							
	Фамилия varchar(20) NOT NULL																							
	Имя varchar(20) NOT NULL																							
	Отчество varchar(20) NULL																							
2.	<p>Для базы данных по заданной схеме написать триггер, осуществляющие удаление группы, если в ней не осталось студентов.</p> <div><table><tr><th colspan="2">Группа</th></tr><tr><td>PK</td><td><u>идГр int NOT NULL</u></td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>кол_во_студ int NOT NULL</td></tr></table><table><tr><th colspan="2">Студент</th></tr><tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></table></div>	Группа		PK	<u>идГр int NOT NULL</u>		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		кол_во_студ int NOT NULL	Студент		PK	<u>ид_Студ int NOT NULL</u>	FK1	идГр int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	ОПК-6.У.1
Группа																								
PK	<u>идГр int NOT NULL</u>																							
	год_поступления int NOT NULL																							
	НомерГр varchar(10) NOT NULL																							
	кол_во_студ int NOT NULL																							
Студент																								
PK	<u>ид_Студ int NOT NULL</u>																							
FK1	идГр int NOT NULL																							
	Фамилия varchar(20) NOT NULL																							
	Имя varchar(20) NOT NULL																							
	Отчество varchar(20) NULL																							
3.	Добавить пользователя Perfect в базу данных «uni» дать ему право на изменение только таблицы «Группа», а остальных таблиц права на чтение.	ОПК-6.У.1																						
4.	Для базы данных по схеме представленной ниже, определить и обосновать уровень изоляции транзакций, необходимых для работы приложения а) деканата б) личного кабинета студента	ОПК-6.У.1																						

	<div><div><div><div><div>Группа</div><div><div>PK</div><div><u>идГр int NOT NULL</u></div></div><div>год_поступления int NOT NULL</div><div>НомерГр varchar(10) NOT NULL</div></div></div><div><div><div>Кружок</div><div><div>PK</div><div><u>ид_кружка int NOT NULL</u></div></div><div>FK1 ид_руководителя int NOT NULL</div><div>Название varchar(50) NOT NULL</div></div></div><div><div><div>Преподаватель</div><div><div>PK</div><div><u>ид_препод int NOT NULL</u></div></div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Студент</div><div><div>PK</div><div><u>ид_Студ int NOT NULL</u></div></div><div>FK1 идГр int NOT NULL</div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Кружок_студента</div><div><div>PK,FK1</div><div><u>ид_Студ int NOT NULL</u></div></div><div><div>PK,FK2</div><div><u>ид_кружка int NOT NULL</u></div></div><div>Дата вступления date NULL</div></div></div></div></div>	
5.	Для предметной области «учебный план» (дисциплины, кафедры, знания и умения полученные в результате освоения дисциплин) предложить и обосновать выбор модели данных СУБД. Нарисовать примерную схему базы данных.	ОПК-6.У.1
6.	Для предметной области «3D-моделлер» (простейшие 3D-объекты, пользовательские объекты, комбинации) предложить и обосновать выбор модели данных СУБД. Нарисовать примерную схему базы данных.	ОПК-6.У.1
7.	Для предметной области «составление тестов. тест с управляемой последовательностью вопросов» (вопрос ,варианты ответа, условие выбора следующего вопроса) предложить и обосновать выбор модели данных СУБД. Нарисовать примерную схему базы данных.	ОПК-6.У.1
8.	Для реляционной базы со схемой представленной ниже, создать соответствующую ей (по данным) документную базу данных	ОПК-6.У.1

	<div><div><div><div><div>Группа</div><div><div>PK</div><div><u>идГр int NOT NULL</u></div></div><div>год_поступления int NOT NULL</div><div>НомерГр varchar(10) NOT NULL</div></div></div><div><div><div>Кружок</div><div><div>PK</div><div><u>ид_кружка int NOT NULL</u></div></div><div>FK1 ид_руководителя int NOT NULL</div><div>Название varchar(50) NOT NULL</div></div></div><div><div><div>Преподаватель</div><div><div>PK</div><div><u>ид_препод int NOT NULL</u></div></div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Студент</div><div><div>PK</div><div><u>ид_Студ int NOT NULL</u></div></div><div>FK1 идГр int NOT NULL</div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Кружок_студента</div><div><div>PK,FK1</div><div><u>ид_Студ int NOT NULL</u></div></div><div><div>PK,FK2</div><div><u>ид_кружка int NOT NULL</u></div></div><div>Дата вступления date NULL</div></div></div></div></div>	
9.	<div>Для реляционной базы со схемой ниже, создать соответствующую ей (по данным) базу данных «ключ-значение»</div> <div><div><div><div><div>Группа</div><div><div>PK</div><div><u>идГр int NOT NULL</u></div></div><div>год_поступления int NOT NULL</div><div>НомерГр varchar(10) NOT NULL</div></div></div><div><div><div>Кружок</div><div><div>PK</div><div><u>ид_кружка int NOT NULL</u></div></div><div>FK1 ид_руководителя int NOT NULL</div><div>Название varchar(50) NOT NULL</div></div></div><div><div><div>Преподаватель</div><div><div>PK</div><div><u>ид_препод int NOT NULL</u></div></div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Студент</div><div><div>PK</div><div><u>ид_Студ int NOT NULL</u></div></div><div>FK1 идГр int NOT NULL</div><div>Фамилия varchar(20) NOT NULL</div><div>Имя varchar(20) NOT NULL</div><div>Отчество varchar(20) NULL</div></div></div><div><div><div>Кружок_студента</div><div><div>PK,FK1</div><div><u>ид_Студ int NOT NULL</u></div></div><div><div>PK,FK2</div><div><u>ид_кружка int NOT NULL</u></div></div><div>Дата вступления date NULL</div></div></div></div></div>	ОПК-6.У.1
10.	<div>Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт, добавляющий каждому студенту свойство количество работ с актуальными данными.</div>	ОПК-6.У.1
11.	<div>Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт создания /добавления данных для MongoDB</div>	ОПК-6.У.1

12.	Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт выбирающий студентов, сдавших наибольшее количество работ по дисциплине «Проектирование баз данных»	ОПК-6.У.1																																								
13.	Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт удаляющий все работы, прикреплённые в позапрошлом году	ОПК-6.У.1																																								
14.	Для объектно-реляционной базы данных на схеме представленной ниже, Написать код добавляющий унаследованную таблицу «аспирант» <pre>graph TD     Группа[Группа] -- "1:M" --&gt; Студент[Студент]     Преподаватель[Преподаватель] -- "1:M" --&gt; Студент     Преподаватель -- "1:M" --&gt; Аспирант[Аспирант]     Студент -- "1:M" --&gt; Аспирант</pre> <table border="1"><thead><tr><th colspan="2">Группа</th></tr></thead><tbody><tr><td>PK</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>год_поступления int NOT NULL</td></tr><tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr><tr><td></td><td>кол_во_студ int NOT NULL</td></tr></tbody></table> <table border="1"><thead><tr><th colspan="2">Преподаватель</th></tr></thead><tbody><tr><td>PK</td><td>ид_препод int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></tbody></table> <table border="1"><thead><tr><th colspan="2">Студент</th></tr></thead><tbody><tr><td>PK</td><td>ид_Студ int NOT NULL</td></tr><tr><td>FK1</td><td>идГр int NOT NULL</td></tr><tr><td></td><td>Фамилия varchar(20) NOT NULL</td></tr><tr><td></td><td>Имя varchar(20) NOT NULL</td></tr><tr><td></td><td>Отчество varchar(20) NULL</td></tr></tbody></table> <table border="1"><thead><tr><th colspan="2">Аспирант</th></tr></thead><tbody><tr><td>PK</td><td>ид_Студ int NOT NULL</td></tr><tr><td>FK1</td><td>ид_руководителя int NOT NULL</td></tr><tr><td></td><td>Научная специальность varchar(10) NOT NULL</td></tr></tbody></table>	Группа		PK	идГр int NOT NULL		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL		кол_во_студ int NOT NULL	Преподаватель		PK	ид_препод int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Студент		PK	ид_Студ int NOT NULL	FK1	идГр int NOT NULL		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	Аспирант		PK	ид_Студ int NOT NULL	FK1	ид_руководителя int NOT NULL		Научная специальность varchar(10) NOT NULL	ОПК-6.У.1
Группа																																										
PK	идГр int NOT NULL																																									
	год_поступления int NOT NULL																																									
	НомерГр varchar(10) NOT NULL																																									
	кол_во_студ int NOT NULL																																									
Преподаватель																																										
PK	ид_препод int NOT NULL																																									
	Фамилия varchar(20) NOT NULL																																									
	Имя varchar(20) NOT NULL																																									
	Отчество varchar(20) NULL																																									
Студент																																										
PK	ид_Студ int NOT NULL																																									
FK1	идГр int NOT NULL																																									
	Фамилия varchar(20) NOT NULL																																									
	Имя varchar(20) NOT NULL																																									
	Отчество varchar(20) NULL																																									
Аспирант																																										
PK	ид_Студ int NOT NULL																																									
FK1	ид_руководителя int NOT NULL																																									
	Научная специальность varchar(10) NOT NULL																																									
15.	Для объектно-реляционной базы данных (схема представлена ниже), определить другие примеры наследования	ОПК-6.У.1																																								

16.	<p>Для объектно-реляционной базы данных (схема представлена ниже), предложить пользовательский оператор и реализовать его для PostgreSQL</p>	ОПК-6.У.1

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	<p>Инструкция: выберите один правильный ответ</p> <p>Какие функции выполняет система управления базами данных?</p> <ol style="list-style-type: none"> <li>1. Хранение данных</li> <li>2. Обработка данных</li> <li>3. Управление доступом к данным</li> <li>4. Резервное копирование данных</li> </ol>	ОПК-5.3.1
2	<p>Инструкция: выберите несколько правильных ответов</p> <p>Какие методы защиты информации используются в системах</p>	ОПК-5.3.1

	<p>управления базами данных?</p> <ol style="list-style-type: none"> <li>1. Шифрование данных</li> <li>2. Аутентификация пользователей</li> <li>3. Резервное копирование данных</li> <li>4. Разграничение доступа к данным</li> <li>5. Антивирусная защита</li> </ol>	
3	<p>Инструкция: установите соответствия</p> <p>Установите соответствие между этапами жизненного цикла программного обеспечения и их описаниями.</p> <p>Этапы:</p> <ol style="list-style-type: none"> <li>1. Анализ требований</li> <li>2. Проектирование</li> <li>3. Реализация</li> <li>4. Тестирование</li> <li>5. Внедрение</li> <li>6. Сопровождение</li> </ol> <p>Описание:</p> <ol style="list-style-type: none"> <li>1. Определение функциональных и не функциональных требований к системе</li> <li>2. Создание архитектуры системы и разработка спецификаций</li> <li>3. Кодирование, компиляция и интеграция компонентов системы</li> <li>4. Проверка работоспособности системы и поиск ошибок</li> <li>5. Установка и настройка системы в рабочей среде</li> <li>6. Поддержка и обновление системы после ее внедрения</li> </ol>	ОПК-5.3.1
4	<p>Инструкция: установите последовательность</p> <p>Установите последовательность этапов жизненного цикла программного обеспечения.</p> <ol style="list-style-type: none"> <li>1. Анализ требований</li> <li>2. Проектирование</li> <li>3. Тестирование</li> <li>4. Реализация</li> <li>5. Внедрение</li> <li>6. Сопровождение</li> </ol>	ОПК-5.3.1
5	<p>Инструкция: Каковы основные принципы работы системы управления базами данных?</p>	ОПК-5.3.1
6	<p>Инструкция: выберите один правильный ответ</p> <p>Какая модель данных основана на концепции таблиц, каждая из которых представляет собой набор отношений между сущностями?</p> <ol style="list-style-type: none"> <li>1. Объектно-ориентированная</li> <li>2. Реляционная</li> <li>3. Графовая</li> <li>4. Сетевая</li> <li>5. Хранилища данных</li> </ol>	ОПК-6.3.1
7	<p>Инструкция: выберите один правильный ответ</p> <p>Дана сводная таблица по работам студентов</p>	ОПК-6.У.1

	<div>Работы</div> <table><tr><th>Студент</th><th>Дисциплина</th><th>Работа</th><th>Балл</th></tr><tr><td>Иванов</td><td>ООП</td><td>1</td><td>8</td></tr><tr><td>Петров</td><td>ООП</td><td>2</td><td>6</td></tr><tr><td>Сидоров</td><td>ПБД</td><td>1</td><td>4</td></tr><tr><td>Иванов</td><td>ПБД</td><td>1</td><td>Null</td></tr></table> <p>Какое число вернет запрос ниже? SELECT avg(Балл) from Работы</p> <div><div>1. 6</div><div>2. 4,5</div><div>3. Null</div><div>4. Ничего не выведет, выдаст ошибку, т.к. агрегатные функции не работают с null-значениями</div></div>	Студент	Дисциплина	Работа	Балл	Иванов	ООП	1	8	Петров	ООП	2	6	Сидоров	ПБД	1	4	Иванов	ПБД	1	Null	
Студент	Дисциплина	Работа	Балл																			
Иванов	ООП	1	8																			
Петров	ООП	2	6																			
Сидоров	ПБД	1	4																			
Иванов	ПБД	1	Null																			
8	<p>Инструкция: выберите несколько правильных ответов</p> <p>Какой тип данных является числовым типом с фиксированной запятой?</p> <div><div>1. Numeric</div><div>Decimal</div></div>	ОПК-6.3.1																				
9	<p>Инструкция: выберите несколько правильных ответов</p> <p>В операторе создания таблиц можно:</p> <div><div>1. указать способы поддержания ссылочной целостности</div><div>2. указать внешний ключ таблицы</div><div>3. указать значение по умолчанию для столбца</div><div>4. указать вторичный ключ в таблицы</div><div>5. создать домен столбца</div><div>6. назначить псевдоним по умолчанию для таблицы</div></div>	ОПК-6.У.1																				
10	<p>Инструкция: установите соответствия</p> <p>Установите соответствие между названием языка и его описанием.</p> <p>Название:</p> <div><div>1. DDL</div><div>2. DML</div></div> <p>Описание:</p> <div><div>1. Язык определения данных, используется для определения, изменения и управления структурой базы данных, включая создание и изменение таблиц, представлений, индексов и других объектов базы данных</div></div> <p>Язык манипулирования данными, используется для манипулирования данными, хранящимися в базе данных, включая добавление, изменение и удаление записей в таблицах</p>	ОПК-6.3.1																				
11	<p>Инструкция: установите соответствия</p> <p>Дана сводная таблица по работам студентов</p> <div>Работы</div> <table><tr><th>Студент</th><th>Дисциплина</th><th>Работа</th><th>Балл</th></tr><tr><td>Иванов</td><td>ООП</td><td>1</td><td>8</td></tr><tr><td>Петров</td><td>ООП</td><td>2</td><td>6</td></tr><tr><td>Сидоров</td><td>ПБД</td><td>1</td><td>4</td></tr><tr><td>Иванов</td><td>ПБД</td><td>1</td><td>Null</td></tr></table> <p>Установите соответствие между запросом и результатом его выполнения.</p> <p>Запрос:</p> <div><div>1. SELECT Студент from Работы</div><div>2. SELECT distinct(Студент) from Работы</div></div> <p>Результат выполнения запроса:</p>	Студент	Дисциплина	Работа	Балл	Иванов	ООП	1	8	Петров	ООП	2	6	Сидоров	ПБД	1	4	Иванов	ПБД	1	Null	ОПК-6.У.1
Студент	Дисциплина	Работа	Балл																			
Иванов	ООП	1	8																			
Петров	ООП	2	6																			
Сидоров	ПБД	1	4																			
Иванов	ПБД	1	Null																			

	<div><div>1.</div><div><table><tr><td>Студент</td></tr><tr><td>Иванов</td></tr><tr><td>Петров</td></tr><tr><td>Сидоров</td></tr><tr><td>Иванов</td></tr></table></div></div> <div><div>2.</div><div><table><tr><td>Студент</td></tr><tr><td>Иванов</td></tr><tr><td>Петров</td></tr><tr><td>Сидоров</td></tr></table></div></div>	Студент	Иванов	Петров	Сидоров	Иванов	Студент	Иванов	Петров	Сидоров												
Студент																						
Иванов																						
Петров																						
Сидоров																						
Иванов																						
Студент																						
Иванов																						
Петров																						
Сидоров																						
12	<p>Инструкция: установите последовательность шагов при разработке базы данных:</p> <div><div>1.</div>Определение требований к базе данных</div> <div><div>2.</div>Проектирование логической структуры базы данных</div> <div><div>3.</div>Проектирование физической структуры базы данных</div> <div><div>4.</div>Тестирование и внедрение базы данных</div> <p>Разработка пользовательского интерфейса</p>	ОПК-6.3.1																				
13	<p>Инструкция: установите последовательность. Перечислите порядок выполнения операторов в SQL запросе:</p> <div><div>1.</div>Select</div> <div><div>2.</div>From</div> <div><div>3.</div>Where</div> <div><div>4.</div>Group by</div>	ОПК-6.У.1																				
14	<p>Инструкция: Дайте определение слову «таблица»</p>	ОПК-6.3.1																				
15	<p>Инструкция: Дана сводная таблица по работам студентов</p> <div><div>Работы</div><table><tr><td>Студент</td><td>Дисциплина</td><td>Работа</td><td>Балл</td></tr><tr><td>Иванов</td><td>ООП</td><td>1</td><td>8</td></tr><tr><td>Петров</td><td>ООП</td><td>2</td><td>6</td></tr><tr><td>Сидоров</td><td>ПБД</td><td>1</td><td>4</td></tr><tr><td>Иванов</td><td>ПБД</td><td>1</td><td>Null</td></tr></table></div> <p>Напишите запрос для добавления данных в изображенную таблицу?</p>	Студент	Дисциплина	Работа	Балл	Иванов	ООП	1	8	Петров	ООП	2	6	Сидоров	ПБД	1	4	Иванов	ПБД	1	Null	ОПК-6.У.1
Студент	Дисциплина	Работа	Балл																			
Иванов	ООП	1	8																			
Петров	ООП	2	6																			
Сидоров	ПБД	1	4																			
Иванов	ПБД	1	Null																			

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Проектирование взаимодействия базы данных и приложения (6 семестр)



1.1	<p>Создать для базы данных программы для рисования графов приложение с двумя способами подключения базы данных:</p> <p>1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Вершины, название/текст которых содержит слово «конец», но не заканчивается им</li> <li>б. Вершины, у которых нет исходящих ребер</li> <li>в. Графы, в которых есть пара вершин, связанных ребрами в обе стороны</li> <li>г. Ширина графа в пикселях (от максимальной сумма координаты по горизонтали с шириной отнять минимальную левую координату)</li> <li>д. Пользователь- автор, графов с максимальным количеством вершин</li> <li>е. Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам</li> <li>ж. Вершина, нет входящих ребер от вершины 2</li> </ul>
1.2	<p>Создать для базы данных Садоводства приложение с двумя способами подключения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. номера участков владельцев с отчеством, заканчивающимся на «вна»</li> <li>б. участки, на которых зарегистрировано более1 постройки</li> <li>в. тип взносов, которые пока никто не оплатил</li> <li>г. Владелец (владельцы) участка максимальной площади</li> <li>д. Владельцы максимального количества участков</li> <li>е. Участки, на которых есть постройки всех типов</li> <li>ж. Участки, на которых нет бань</li> </ul>
1.3	<p>Создать для базы данных ТСЖ приложение с двумя способами подключения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. квартиры владельцев, отчества которых заканчиваются на 'ич'</li> <li>б. владельцев, у которых есть квартиры на разных этажах</li> <li>в. квартиры, в которых не записано ни одного показания счетчика газа</li> <li>г. владельцы квартир с минимальной жилой площадью</li> <li>д. этаж, на котором меньше всего квартир находится в собственности</li> <li>е. квартиры, в которых стоят все типы счетчиков</li> <li>ж. владелец, у которого нет трехкомнатных квартир</li> </ul>

1.4	<p>Создать для базы данных парка приложение с двумя способами подсоединения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView, с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. аллеи, на которых встречаются разные виды кленов (клен в названии)</li> <li>б. аллеи, состоящие из разных пород деревьев</li> <li>в. породы, не высаженные в парке</li> <li>г. дерево, которое было посажено раньше всех</li> <li>д. порода, деревьев которой меньше всего</li> <li>е. аллея, на которой растут деревья всех пород, которые есть в парке</li> <li>ж. аллея, деревья на которой не высаживались в прошлом году</li> </ul>
1.5	<p>Создать для базы данных расписания экзаменов/зачетов приложение с двумя способами подсоединения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView, с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. аудитории, которых проходят экзамены по дисциплинам, имеющим в названии слова «базы данных»</li> <li>б. аудитории, где в один день проходит несколько экзаменов</li> <li>в. преподаватели, не принимающие экзаменов</li> <li>г. дисциплины самого последнего экзамена</li> <li>д. аудитории, в которых проходит больше всего экзаменов</li> <li>е. дисциплина, по которой есть экзамены у всех групп</li> <li>ж. преподаватель, не принимающий экзамены у группы Z9431</li> </ul>

1.6	<p>Создать для базы данных поликлиники приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. пациенты, приходившие на любые процедуры, связанные с электрофорезом</li> <li>б. пациент, приходивший к одному врачу и на прием и на процедуру</li> <li>в. пациенты, не ходившие в поликлинику</li> <li>г. процедуры с наименьшей стоимостью</li> <li>д. пациент, ходивший к наибольшему количеству врачей</li> <li>е. пациент, ходивший на все процедуры</li> <li>ж. пациент, не ходивший на процедуры к Иванову Ивану Ивановичу</li> </ul>
1.7	<p>Создать для базы данных личного кабинета студента приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. преподаватели, аз которыми закреплены дисциплины, начинающиеся со слова «автоматизирован»</li> <li>б. дисциплина, по которой есть и лабораторные и курсовая работа</li> <li>в. студенты, не прикрепившие ни одной работы</li> <li>г. студент, сдавший курсовую раньше всех</li> <li>д. студент, прикрепивший в этом месяце наибольшее число работ</li> <li>е. дисциплина, по которой есть все типы работ (КР, ЛР, практические)</li> <li>ж. преподаватель, которому не прикрепляли отчетов по курсовому проекту или работе</li> </ul>
1.8	<p>Создать для базы данных костюмерной театра приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. спектакли, в которых используются костюмы, имеющие в названии слово шут</li> <li>б. костюм, в котором есть и куртка и штаны</li> <li>в. роль, на которую пока нет костюма</li> <li>г. автор, разработавший самый старый из костюмов</li> <li>д. спектаклю, к которому разработано наибольшее число костюмов</li> <li>е. костюм, в котором есть все типы деталей</li> <li>ж. автор, не разрабатывавший костюмы к «Золушке»</li> </ul>

1.9	<p>Создать для базы данных охраняемых парковок приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все парковки, расположенные на линиях (не улицах или проспектах)(улица в адресе содержит «линия»)</li> <li>б. владелец машины, у которого более одного места под машину</li> <li>в. машины, которые неизвестно когда уедут</li> <li>г. владелец машин, заезжавший раньше всех</li> <li>д. владелец машины, останавливавшийся на минимальном числе парковок</li> <li>е. машина, которая стояла на всех парковках Московского проспекта</li> <li>ж. владелец, не парковавшийся на Невском проспекте</li> </ul>
1.10	<p>Создать для базы данных спортзала приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. виды спорта, на которые есть без лимитные абонементы (со словом без лимит в названии типа)</li> <li>б. вид спорта, по которому занятия ведут различные тренеры</li> <li>в. вид спорта, по которому не куплено ни одного абонемента</li> <li>г. тренеры, ведущие занятия по виду спорта с минимальным число занятий в абонементе</li> <li>д. тренеры, которые ведут занятия по максимальному количеству видов спорта</li> <li>е. тренеры, не ведущие фитнес</li> <li>ж. абонемент, на все виды спорта, связанные с единоборствами, заканчивающимися на «до»</li> </ul>
1.11	<p>Создать для базы данных магазина обоев приложение с двумя способами подсоединения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. коллекция обоев, начинающаяся со слова Элегия</li> <li>б. производитель, который производит как бумажные, так и флизелиновые обои</li> <li>в. текстура, обоев с которой нет</li> <li>г. производитель с самой большой шириной рулона</li> <li>д. производитель обоев с максимальным количеством коллекций</li> <li>е. производитель, который производит обои из всех материалов</li> <li>ж. производитель, у которого нет обоев дороже 3000 рублей</li> </ul>

1.12	<p>Создать для базы данных расписания пригородных электричек приложение с двумя способами подсоединения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все маршруты, проходящие через остановку , в названии которой есть слово село (царское село, детское село)</li> <li>б. свежестроенная станция, к которой пока не ведет ни один маршрут</li> <li>в. маршрут, проходящий через Купчино и Шушары</li> <li>г. самая поздняя электричка, прибывающая на витебский вокзал</li> <li>д. маршруты с наибольшим количеством электричек</li> <li>е. маршрут, на котором нет электричек с режимом отправления по выходным.</li> <li>ж. маршрут, по которому ходят электрички со всеми режимами движения</li> </ul>
1.13	<p>Создать для вакансий базы данных приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. вакансии, имеющие в названии SQL</li> <li>б. подозрительные вакансии, где не прописаны обязанности</li> <li>в. работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора</li> <li>г. вакансия с наименьшей зарплатой</li> <li>д. вакансии с минимальным количеством обязательных требований</li> <li>е. требование, присутствующее во всех вакансиях на программиста</li> <li>ж. вакансии, в которых нет требования к опыту работы</li> </ul>
1.14	<p>Создать для базы данных калькулятора бюджета физического лица приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все расходы категорий, которые относятся к спорту (содержат слово спорт)</li> <li>б. месяц, в котором были разные статьи дохода</li> <li>в. статья доходов, по которой доходов не было</li> <li>г. категория, по которой наибольшие расходы в текущем году</li> <li>д. категория, по которой не было расходов в январе</li> <li>е. месяц, в котором были траты всех категорий расхода</li> <li>ж. месяц, в котором были траты максимального количества статей</li> </ul>

1.15	<p>Создать для базы данных книги контактов приложение с двумя способами подсоединения базы данных: 1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. друзья, у которых номер телефона начинается на 8-911,</li> <li>б. Люди без меток,</li> <li>в. люди, которые относятся к соседям и коллегам одновременно</li> <li>г. самые старые люди среди контактов</li> <li>д. месяц, когда есть дни рождения у соседей, но нет у семьи</li> <li>е. метки, к которым относится максимальное количество людей</li> <li>ж. месяц, в котором есть дни рождения у людей со всеми метками</li> </ul>
1.16	<p>Создать для базы данных вузов для абитуриентов приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. направленности, в которых есть слово автоматизированный</li> <li>б. факультет, не принимающий ни на одно направление</li> <li>в. направление, на которое надо сдавать историю и обществознание</li> <li>г. факультет вуза, принимающий на наименьшее количество направлений</li> <li>д. город, в котором есть все направления</li> <li>е. вуз, с самым ранним началом приемной компании</li> <li>ж. направление, на которое не надо сдавать ЕГЭ по информатике</li> </ul>
1.17	<p>Создать для базы данных школьных экскурсий приложение с двумя способами подсоединения базы данных: 1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. экскурсии в музеи (слово музей в любом месте названия)</li> <li>б. экскурсии, относящиеся к литературе и истории</li> <li>в. учитель, который не отвечает за экскурсии</li> <li>г. учащиеся, которые не ездили в музей истории религии</li> <li>д. экскурсия, собравшая наибольшее число участников</li> <li>е. самые дорогие экскурсии</li> <li>ж. учащийся, который был на всех экскурсиях</li> </ul>

1.18	<p>Создать для базы данных собачьего питомника приложение с двумя способами подсоединения базы данных:</p> <p>1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. породы, названия которых содержат слово «шпиц», но не начинается с него</li> <li>б. собаки без медалей</li> <li>в. собаки, у родителей которых один и тот же владелец</li> <li>г. владелец, у которого есть собаки всех пород</li> <li>д. владелец, у которого есть йоркширские терьеры, но нет мастиффов</li> <li>е. порода, собак которой меньше всего</li> <li>ж. владельцы самых старых собак</li> </ul>
1.19	<p>Создать для базы данных службы доставки приложение с двумя способами подсоединения базы данных:</p> <p>1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все посылки, отправляемые на улицу, в названии которой есть окончание «ая», но это не единственные буквы в нем</li> <li>б. улица, на которую никогда не доставляли посылки</li> <li>в. контактное лицо, получавшее посылки по разным адресам</li> <li>г. адреса, по которым доставили больше всего посылок</li> <li>д. посылка с самым маленьким весом</li> <li>е. контактное лицо, получавшее посылки от всех фирм</li> <li>ж. контактное лицо, никогда не получавшее посылок в январе</li> </ul>
1.20	<p>Создать для базы данных туристического путеводителя приложение с двумя способами подсоединения базы данных:</p> <p>1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение</p>

	<p>третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. достопримечательности, в которых есть слово «мать», но с него название не начинается</li> <li>б. город без достопримечательностей</li> <li>в. улица, на которой есть и памятники, и архитектурные комплексы</li> <li>г. город, в котором нет природных комплексов</li> <li>д. улица, на которой больше всего памятников</li> <li>е. города с самыми старыми достопримечательностями</li> <li>ж. улица, на которой есть достопримечательности всех типов</li> </ul>
1.21	<p>Создать для базы данных метрополитена приложение с двумя способами подсоединения базы данных:</p> <p>1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. станции, в названии которых есть слово «площадь», но оно на него не заканчивается</li> <li>а. линии без станций (в проекте)</li> <li>б. пересадочные станции с линии 1 на линию 2</li> <li>в. линия с самым большим временем проезда</li> <li>г. линия, на которой собраны все станции, открываются раньше 7 утра</li> <li>д. станция с самым ранним открытием</li> <li>е. линия на которой нет перегона больше 3 минут</li> </ul>
1.22	<p>Создать для базы данных медпункта организации приложение с двумя способами подсоединения базы данных:</p> <p>1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все сотрудники, женского пола (отчество, оканчивается на «на»)</li> <li>б. сотрудники, никогда не обращавшиеся в медпункт с жалобами</li> <li>в. сотрудники организации, делавшие за последний год прививки от столбняка и кори</li> <li>г. сотрудник, последним сделавший флюорографию</li> <li>д. сотрудник, у которого сделаны все типы прививок</li> <li>е. сотрудник с самым большим количеством жалоб</li> <li>ж. сотрудник, не делавший прививки от энцефалита</li> </ul>
1.23	<p>Создать для базы данных по материальной ответственности приложение с двумя способами подсоединения базы данных:</p> <p>1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p>



	<p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. любые помещения, где стоит компьютерная мебель (категория мебель и слово «комп»)</li> <li>б. помещение, в котором нет никакого имущества</li> <li>в. материально ответственное лицо, у которого подотчетна и компьютерная техника и специальное оборудование</li> <li>г. имущество, списанное последним</li> <li>д. помещение, в котором есть имущество всех категорий</li> <li>е. мат. ответственное лицо, ответственное за наибольшее количество имущества</li> <li>ж. помещение, в котором нет компьютерной техники</li> </ul>
1.24	<p>Создать для базы данных календаря корпоративных мероприятий приложение с двумя способами подсоединения базы данных:</p> <p>1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Все мероприятия, текущего года, в названии которых есть слово конференция, но с него с названия не начинаются</li> <li>б. Комната, в которой не проходят мероприятия</li> <li>в. Мероприятия, в которых участвуют сотрудники «подразделения 1» и «подразделения 2»</li> <li>г. Участники самых поздно заканчивающихся мероприятий</li> <li>д. Событие, с самым большим количеством пришедших на него участников.(участники на мероприятие могут быть приглашены и при этом могут придти/не придти).</li> <li>е. Мероприятие, в котором обязательно присутствие всех начальников отделов</li> <li>ж. Сотрудник, не посещавший мероприятий в июле прошлого года</li> </ul>
1.25	<p>Создать для базы данных домашних обязанностей приложение с двумя способами подсоединения базы данных:</p> <p>1)Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView ,с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Люди, выполняющие обязанности, имеющие в названии слово уборка, но не заканчивающиеся на него</li> <li>б. Обязанность, которую еще не распределили по выполняющим людям</li> <li>в. Обязанность, которую выполняет более одного человека</li> <li>г. Ежемесячная обязанность, которая выполняется первой по дате (дню)</li> <li>д. Категория, обязанности из которой выполняют все</li> <li>е. Категория, в которой меньше всего обязанностей</li> <li>ж. Человек, который не выполняет обязанности из категории «оплата счетов»</li> </ul>
1.26	<p>Создать для базы данных рабочих задач приложение с двумя способами подсоединения базы данных:</p>

	<p>1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DataGridView, с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.</p> <p>2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Задачи, в названии которых содержится слово «интеграция», но оно не последнее</li> <li>б. Проект без задач</li> <li>в. Задача, относящаяся к 2 различным проектам</li> <li>г. Задача с самым поздним сроком окончания</li> <li>д. Проект с самым большим количеством задач</li> <li>е. Человек, выполняющий задачи всех приоритетов</li> <li>ж. Человек, у которого нет незавершенных задач</li> </ul>
<b>2</b>	<b>Триггеры. Обеспечение активной целостности данных базы данных (7 семестр)</b>
2.1	<p>Реализовать для базы данных программы для рисования графов триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Вершины, название/текст которых содержит слово «конец», но не заканчивается им</li> <li>б. Вершины, у которых нет исходящих ребер</li> <li>в. Графы, в которых есть пара вершин, связанных ребрами в обе стороны</li> <li>г. Ширина графа в пикселях (от максимальной сумма координаты по горизонтали с шириной отнять минимальную левую координату)</li> <li>д. Пользователь- автор, графов с максимальных количеством вершин</li> <li>е. Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам</li> <li>ж. Вершина, нет входящих ребер от вершины 2</li> </ul>
2.2	<p>Реализовать для базы данных Садоводства триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. номера участков владельцев с отчеством, заканчивающимся на «вна»</li> <li>б. участки, на которых зарегистрировано более 1 постройки</li> <li>в. тип взносов, которые пока никто не оплатил</li> <li>г. Владелец (владельцы) участка максимальной площади</li> <li>д. Владельцы максимального количества участков</li> <li>е. Участки, на которых есть постройки всех типов</li> <li>ж. Участки, на которых нет бань</li> </ul>
2.3	<p>Реализовать для базы данных ТСЖ триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. квартиры владельцев, отчества которых заканчиваются на 'ич'</li> </ul>

	<ul style="list-style-type: none"> <li>б. владельцев, у которых есть квартиры на разных этажах</li> <li>в. квартиры, в которых не записано ни одного показания счетчика газа</li> <li>г. владельцы квартир с минимальной жилой площадью</li> <li>д. этаж, на котором меньше всего квартир находится в собственности</li> <li>е. квартиры, в которых стоят все типы счетчиков</li> <li>ж. владелец, у которого нет трехкомнатных квартир</li> </ul>
2.4	<p>Реализовать для базы данных парка триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. аллеи, на которых встречаются разные виды кленов (клен в названии)</li> <li>б. аллеи, состоящие из разных пород деревьев</li> <li>в. породы, не высаженные в парке</li> <li>г. дерево, которое было посажено раньше всех</li> <li>д. порода, деревьев которой меньше всего</li> <li>е. аллея, на которой растут деревья всех пород, которые есть в парке</li> <li>ж. аллея, деревья на которой не высаживались в прошлом году</li> </ul>
2.5	<p>Реализовать для базы данных расписания экзаменов/зачетов триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. аудитории, которых проходят экзамены по дисциплинам, имеющим в названии слова «базы данных»</li> <li>б. аудитории, где в один день проходит несколько экзаменов</li> <li>в. преподаватели, не принимающие экзаменов</li> <li>г. дисциплины самого последнего экзамена</li> <li>д. аудитории, в которых проходит больше всего экзаменов</li> <li>е. дисциплина, по которой есть экзамены у всех групп</li> <li>ж. преподаватель, не принимающий экзамены у группы Z9431</li> </ul>
2.6	<p>Реализовать для базы данных поликлиники триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. пациенты, приходившие на любые процедуры, связанные с электрофорезом</li> <li>б. пациент, приходивший к одному врачу и на прием и на процедуру</li> <li>в. пациенты, не ходившие в поликлинику</li> <li>г. процедуры с наименьшей стоимостью</li> <li>д. пациент, ходивший к наибольшему количеству врачей</li> <li>е. пациент, ходивший на все процедуры</li> <li>ж. пациент, не ходивший на процедуры к Иванову Ивану Ивановичу</li> </ul>
2.7	<p>Реализовать для базы данных личного кабинета студента триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. преподаватели, аз которыми закреплены дисциплины, начинающиеся со слова</li> </ul>

	<p>«автоматизирован»</p> <p>б. дисциплина, по которой есть и лабораторные и курсовая работа</p> <p>в. студенты, не прикрепившие ни одной работы</p> <p>г. студент, сдавший курсовую раньше всех</p> <p>д. студент, прикрепивший в этом месяце наибольшее число работ</p> <p>е. дисциплина, по которой есть все типы работ (КР, ЛР, практические)</p> <p>ж. преподаватель, которому не прикрепляли отчетов по курсовому проекту или работе</p>
2.8	<p>Реализовать для базы данных костюмерной театра триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <p>а. спектакли, в которых используются костюмы, имеющие в названии слово шут</p> <p>б. костюм, в котором есть и куртка и штаны</p> <p>в. роль, на которую пока нет костюма</p> <p>г. автор, разработавший самый старый из костюмов</p> <p>д. спектаклю, к которому разработано наибольшее число костюмов</p> <p>е. костюм, в котором есть все типы деталей</p> <p>ж. автор, не разрабатывавший костюмы к «Золушке»</p>
2.9	<p>Реализовать для базы данных охраняемых парковок триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <p>а. все парковки, расположенные на линиях (не улицах или проспектах)(улица в адресе содержит «линия»)</p> <p>б. владелец машины, у которого более одного места под машину</p> <p>в. машины, которые неизвестно когда уедут</p> <p>г. владелец машин, заезжавший раньше всех</p> <p>д. владелец машины, останавливавшийся на минимальном числе парковок</p> <p>е. машина, которая стояла на всех парковках Московского проспекта</p> <p>ж. владелец, не парковавшийся на Невском проспекте</p>
2.10	<p>Реализовать для базы данных спортзала триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <p>з. виды спорта, на которые есть безлимитные абонементы (со словом безлимит в названии типа)</p> <p>и. вид спорта, по которому занятия ведут различные тренеры</p> <p>к. вид спорта, по которому не куплено ни одного абонемента</p> <p>л. тренеры, ведущие занятия по виду спорта с минимальным число занятий в абонементе</p> <p>м. тренеры, которые ведут занятия по максимальному количеству видов спорта</p> <p>н. тренеры, не ведущие фитнес</p> <p>о. абонемент, на все виды спорта, связанные с единоборствами, заканчивающимися на «до»</p>
2.11	<p>Реализовать для базы данных магазина обоев триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие</p>

	<p>запросы:</p> <ul style="list-style-type: none"> <li>з. коллекция обоев, начинающаяся со слова Элегия</li> <li>и. производитель, который производит как бумажные, так и флизелиновые обои</li> <li>к. текстура, обоев с которой нет</li> <li>л. производитель с самой большой шириной рулона</li> <li>м. производитель обоев с максимальным количеством коллекций</li> <li>н. производитель, который производит обои из всех материалов</li> <li>о. производитель, у которого нет обоев дороже 3000 рублей</li> </ul>
2.12	<p>Реализовать для базы данных расписания пригородных электричек триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все маршруты, проходящие через остановку , в названии которой есть слово село (царское село, детское село)</li> <li>б. новая станция, к которой пока не ведет ни один маршрут.</li> <li>маршрут, проходящий через Купчино и Шушары</li> <li>г. самая поздняя электричка, прибывающая на витебский вокзал</li> <li>д. маршруты с наибольшим количеством электричек</li> <li>е. маршрут, на котором нет электричек с режимом отправления по выходным.</li> <li>ж. маршрут, по которому ходят электрички со всеми режимами движения</li> </ul>
2.13	<p>Реализовать для вакансий базы данных триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. вакансии, имеющие в названии SQL</li> <li>б. подозрительные вакансии, где не прописаны обязанности</li> <li>в. работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора</li> <li>г. вакансия с наименьшей зарплатой</li> <li>д. вакансии с минимальным количеством обязательных требований</li> <li>е. требование, присутствующее во всех вакансиях на программиста</li> <li>ж. вакансии, в которых нет требования к опыту работы</li> </ul>
2.14	<p>Реализовать для базы данных калькулятора бюджета физического лица триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все расходы категорий, которые относятся к спорту (содержат слово спорт)</li> <li>б. месяц, в котором были разные статьи дохода</li> <li>в. статья доходов, по которой доходов не было</li> <li>г. категория, по которой наибольшие расходы в текущем году</li> <li>д. категория, по которой не было расходов в январе</li> <li>е. месяц, в котором были траты всех категорий расхода</li> <li>ж. месяц, в котором были траты максимального количества статей</li> </ul>
2.15	<p>Реализовать для базы данных книги контактов триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p>

	<p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. друзья, у которых номер телефона начинается на 8-911,</li> <li>б. Люди без меток,</li> <li>в. люди, которые относятся к соседям и коллегам одновременно</li> <li>г. самые старые люди среди контактов</li> <li>д. месяц, когда есть дни рождения у соседей, но нет у семьи</li> <li>е. метки, к которым относится максимальное количество людей</li> <li>ж. месяц, в котором есть дни рождения у людей со всеми метками</li> </ul>
2.16	<p>Реализовать для базы данных вузов для абитуриентов триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. направленности, в которых есть слово автоматизированный</li> <li>б. факультет, не принимающий ни на одно направление</li> <li>в. направление, на которое надо сдавать историю и обществознание</li> <li>г. факультет вуза, принимающий на наименьшее количество направлений</li> <li>д. город, в котором есть все направления</li> <li>е. вуз, с самым ранним началом приемной компании</li> <li>ж. направление, на которое не надо сдавать ЕГЭ по информатике</li> </ul>
2.17	<p>Реализовать для базы данных школьных экскурсий триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. экскурсии в музеи (слово музей в любом месте названия)</li> <li>б. экскурсии, относящиеся к литературе и истории</li> <li>в. учитель, который не отвечает за экскурсии</li> <li>г. учащиеся, которые не ездили в музей истории религии</li> <li>д. экскурсия, собравшая наибольшее число участников</li> <li>е. самые дорогие экскурсии</li> <li>ж. учащийся, который был на всех экскурсиях</li> </ul>
2.18	<p>Реализовать для базы данных собачьего питомника триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. породы, названия которых содержат слово «шпиц», но не начинается с него</li> <li>б. собаки без медалей</li> <li>в. собаки, у родителей которых один и тот же владелец</li> <li>г. владелец, у которого есть собаки всех пород</li> <li>д. владелец, у которого есть йоркширские терьеры, но нет мастиффов</li> <li>е. порода, собак которой меньше всего</li> <li>ж. владельцы самых старых собак</li> </ul>
2.19	<p>Реализовать для базы данных службы доставки триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие</p>

	<p>запросы:</p> <ul style="list-style-type: none"> <li>а. все посылки, отправляемые на улицу, в названии которой есть окончание «ая», но это не единственные буквы в нем</li> <li>б. улица, на которую никогда не доставляли посылки</li> <li>в. контактное лицо, получавшее посылки по разным адресам</li> <li>г. адреса, по которым доставили больше всего посылок</li> <li>д. посылка с самым маленьким весом</li> <li>е. контактное лицо, получавшее посылки от всех фирм</li> <li>ж. контактное лицо, никогда не получавшее посылки в январе</li> </ul>
2.20	<p>Реализовать для базы данных туристического путеводителя триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. достопримечательности, в которых есть слово «мать», но с него название не начинается</li> <li>б. город без достопримечательностей</li> <li>в. улица, на которой есть и памятники, и архитектурные комплексы</li> <li>г. город, в котором нет природных комплексов</li> <li>д. улица, на которой больше всего памятников</li> <li>е. города с самыми старыми достопримечательностями</li> <li>ж. улица, на которой есть достопримечательности всех типов</li> </ul>
2.21	<p>Реализовать для базы данных метрополитена триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. станции, в названии которых есть слово «площадь», но оно на него не заканчивается</li> <li>б. линии без станций (в проекте)</li> <li>в. пересадочные станции с линии 1 на линию 2</li> <li>г. линия с самым большим временем проезда</li> <li>д. линия, на которой собраны все станции, открываются раньше 7 утра</li> <li>е. станция с самым ранним открытием</li> <li>ж. линия на которой нет перегона больше 3 минут</li> </ul>
2.22	<p>Реализовать для базы данных медпункта организации триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. все сотрудники, женского пола (отчество, оканчивается на «на»)</li> <li>б. сотрудники, никогда не обращавшиеся в медпункт с жалобами</li> <li>в. сотрудники организации, делавшие за последний год прививки от столбняка и кори</li> <li>г. сотрудник, последним сделавший флюорографию</li> <li>д. сотрудник, у которого сделаны все типы прививок</li> <li>е. сотрудник с самым большим количеством жалоб</li> <li>ж. сотрудник, не делавший прививки от энцефалита</li> </ul>
2.23	<p>Реализовать для базы данных по материальной ответственности триггеры для всех событий (insert, delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие</p>

	<p>запросы:</p> <ul style="list-style-type: none"> <li>а. любые помещения, где стоит компьютерная мебель (категория мебель и слово «комп»)</li> <li>б. помещение, в котором нет никакого имущества</li> <li>в. материально ответственное лицо, у которого подотчетна и компьютерная техника, специальное оборудование</li> <li>г. имущество, списанное последним</li> <li>д. помещение, в котором есть имущество всех категорий</li> <li>е. мат. ответственное лицо, ответственное за наибольшее количество имущества</li> <li>ж. помещение, в котором нет компьютерной техники</li> </ul>
2.24	<p>Реализовать для базы данных календаря корпоративных мероприятий триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Все мероприятия, текущего года, в названии которых есть слово конференция, но с него с названия не начинаются</li> <li>б. Комната, в которой не проходят мероприятия</li> <li>в. Мероприятия, в которых участвуют сотрудники «подразделения 1» и «подразделения 2»</li> <li>г. Участники самых поздно заканчивающихся мероприятий</li> <li>д. Событие, с самым большим количеством пришедших на него участников.(участники на мероприятие могут быть приглашены и при этом могут придти/не придти).</li> <li>е. Мероприятие, в котором обязательно присутствие всех начальников отделов</li> <li>ж. Сотрудник, не посещавший мероприятий в июле прошлого года</li> </ul>
2.25	<p>Реализовать для базы данных домашних обязанностей триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Люди, выполняющие обязанности, имеющие в названии слово уборка, но не заканчивающиеся на него</li> <li>б. Обязанность, которую еще не распределили по выполняющим людям</li> <li>в. Обязанность, которую выполняет более одного человека</li> <li>г. Ежемесячная обязанность, которая выполняется первой по дате (дню)</li> <li>д. Категория, обязанности из которой выполняют все</li> <li>е. Категория, в которой меньше всего обязанностей</li> <li>ж. Человек, который не выполняет обязанности из категории «оплата счетов»</li> </ul>
2.26	<p>Реализовать для базы данных рабочих задач триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. Задачи, в названии которых содержится слово «интеграция», но оно не последнее</li> <li>б. Проект без задач</li> <li>в. Задача, относящаяся к 2 различным проектам</li> <li>г. Задача с самым поздним сроком окончания</li> <li>д. Проект с самым большим количеством задач</li> <li>е. Человек, выполняющий задачи всех приоритетов</li> <li>ж. Человек, у которого нет незавершенных задач</li> </ul>

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.



### 11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

#### Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

#### Структура предоставления лекционного материала:

Раздел 1. Введение в проектирование баз данных:

Тема 1.1. История развития и причины появления СУБД;

Тема 1.2. Модели данных;

Тема 1.3. Нормализация и денормализация баз данных; Тема 1.4 Роль баз данных в программных системах.

Тема 1.5 Установка и администрирование баз данных

- Раздел 2. Введение в базы данных:

Тема 2.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными;

Тема 2.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных, манипулирования данными;

Тема 2.3. Реляционная алгебра;

- Раздел 3. Введение в язык SQL;

Тема 3.1. Оператор выборки в языке SQL, агрегатные функции;

Тема 3.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов;

Тема 3.3. Использование представлений, управляющих конструкций в языке SQL;

Тема 3.4. Хранимые процедуры, триггеры, обеспечение активной целостности.

Тема 3.5. Управление доступом к базе данных.

- Раздел 4. Объектные и объектно-реляционные базы данных

Тема 4.1. Выбор типа СУБД применительно к особенностям предметной области

Тема 4.2. Объектно-реляционные базы данных

Тема 4.3. Объектные базы данных

- Раздел 5. NoSQL базы данных

Тема 5.1. Базы данных «Ключ-значение»;

Тема 5.2. Документные базы данных;

Тема 5.3. Графовые базы данных;

Тема 5.4. Базы данных «Семейство столбцов».

Лекционный материал сопровождается демонстрацией слайдов.

#### 11.2. Методические указания для обучающихся по участию в семинарах

Основной целью для обучающегося является систематизация и обобщение знаний по изучаемой теме, разделу, формирование умения работать с дополнительными источниками информации, сопоставлять и сравнивать точки зрения, конспектировать прочитанное, высказывать свою точку зрения и т.п. В соответствии с ведущей дидактической целью содержанием семинарских занятий являются узловые, наиболее трудные для понимания и усвоения темы, разделы дисциплины. Спецификой данной формы занятий является совместная работа преподавателя и обучающегося над решением поставленной проблемы, а поиск верного ответа строится на основе чередования индивидуальной и коллективной деятельности.

При подготовке к семинарскому занятию по теме прослушанной лекции необходимо ознакомиться с планом его проведения, с литературой и научными публикациями по теме семинара.

##### Требования к проведению семинаров

Учебным планом не предусмотрено

#### 11.3. Методические указания для обучающихся по прохождению практических занятий *(если предусмотрено учебным планом по данной дисциплине)*

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимися практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

##### Требования к проведению практических занятий

Учебным планом не предусмотрено

#### 11.4. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

### Задание и требования к проведению лабораторных работ

#### Лабораторная работа 1. Разработка физической модели базы данных с учетом декларативной ссылочной целостности

Задание для выполнения в лабораторной работе

1) Создать физическую модель базы данных, находящуюся в третьей нормальной форме в соответствии с заданным вариантом.

2) Описать ссылочную целостность при удалении и обновлении данных. Формат описания:

- Дочерняя таблица;
- Столбцы, составляющие внешний ключ;
- Родительская таблица;
- Наименование ссылочной целостности при удалении;
- Описание действий по поддержанию ссылочной целостности при удалении;
- Наименование ссылочной целостности при обновлении;
- Описание действий по поддержанию ссылочной целостности при обновлении;
- Обоснование выбора типа поддержки ссылочной целостности.

#### Лабораторная работа 2. Создание и модификация базы данных и таблиц базы данных

Задание для выполнения в лабораторной работе

1) Создать базу данных с разработанной в работе 1 физической моделью.

2) Продемонстрировать добавление и удаление столбца командой alter table

#### Лабораторная работа 3. Заполнение таблиц и модификация данных

Задание для выполнения в лабораторной работе

1) Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 2.

В строках, вставляемых в таблицы, должны быть данные как удовлетворяющие, так и не удовлетворяющие условиям запросов, приведенных в варианте задания. (Для демонстрации этого необходимо в отчете создать таблицу, где будет указано задание на запрос, данные удовлетворяющие условиям запроса, данные не удовлетворяющие условиям запроса)

2) Необходимо привести свои примеры использования оператором update и delete и merge

#### Лабораторная работа 4. Разработка SQL запросов: виды соединений и шаблоны

Задание для выполнения в лабораторной работе

1) Реализовать запросы а) .. в), указанные в варианте задания. .

- Запрос на поиск по шаблону (поиск подстроки) выполнить с единственным оператором like

- Запрос на использование одной таблицы несколько раз (псевдонимы).

- Запрос на использование внешних соединений.

Все запросы должны не содержать вложенных запросов или агрегатных функций

#### Лабораторная работа 5. Разработка SQL запросов: запросы с подзапросами

Задание для выполнения в лабораторной работе

1) Реализовать запросы г) .. ж), указанные в варианте задания.

- Запрос с условием на агрегатную функцию

- Запрос с получением агрегатной функции от агрегатной функции

- Запрос на «все» (реляционное деление)

- Запрос на разность 2 запросов.

Один из запросов на максимум/минимум реализовать с помощью директивы all, а второй-без.

Запрос на «все» (реляционное деление) реализовать с помощью 2 not exists

Запросы на разность реализовать в 3 вариантах: Not in, except (MySQL не поддерживает, поэтому только синтаксис), с использованием левого/правого соединения

#### Лабораторная работа 6. Хранимые процедуры

Задание для выполнения в лабораторной работе

Создать в БД хранимые процедуры, реализующие:

— вставку с пополнением справочников (получаем ссылку на внешний ключ по значению данных из родительской таблицы, если данных нет- добавляем в родительскую, затем вставляем в дочернюю);

— удаление с очисткой справочников – удаление данных из родительской таблицы, если после удаления данных из дочерней у строки родительской больше нет зависимых (удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);

— каскадное удаление (удаление всех зависимых данных);

— вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат) (задача- вернуть данные из процедуры/функции);

— формирование статистики во временной таблице. (задача- работа с временными таблицами ).

#### 7 семестр

#### Лабораторная работа 7. Объектно-реляционные базы данных. Проектирование и создание

Задание для выполнения в лабораторной работе

1) Спроектировать физическую модель базы данных, находящуюся в третьей нормальной форме и включающей наследование и хотя бы один пользовательский тип в соответствии с заданным вариантом.

2) Написать соответствующий скрипт создания базы данных

#### Лабораторная работа 8. Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы

Задание для выполнения в лабораторной работе

1) Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 9.

2) Сделать запрос выборки с условием к таблицам предку и потомку. 3) Придумать и создать пользовательский оператор для своей предметной области

4) Придумать и создать пользовательскую агрегатную функцию для своей предметной области

#### Лабораторная работа 9. Разработка документной базы данных

Задание для выполнения в лабораторной работе

1) Спроектировать структуру json файла, соответствующую предметной области по варианту задания.

2) Создать базу MongoDB с тестовыми данными для примера/

#### Лабораторная работа 10. Манипулирование данными в документной базе данных.

Задание для выполнения в лабораторной работе

1) Привести пример обновления и удаления данных из документной базы данных, разработанной в работе 11. (по 2 примера: один из вложенного документа, другой из документа коллекции верхнего уровня)

2) Выполнить запросы на выборку из базы данных в соответствии с вариантом задания. (Запрос на поиск подстроки.

Запрос на использование одного документа несколько раз. Запрос с условием на агрегатную функцию.

Запрос получением агрегатной функции от агрегатной функции. Запрос на разность 2 запросов)

#### Структура и форма отчета о лабораторной работе

В состав отчета должны входить:

- Титульный лист
- Текст задания (с запросами)
- Модель базы данных
- Тестовые данные (при задании связанном с выполнением кода на компьютере)
- Содержательная часть отчета, определяемая заданием
- Скриншоты результата выполнения кода (при задании связанном с выполнением кода на компьютере)

#### Требования к оформлению отчета о лабораторной работе

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде в файле формата pdf, подготовленном в текстовом редакторе, в соответствии с правилами [http://guap.ru/guap/standart/titl\\_main.shtml](http://guap.ru/guap/standart/titl_main.shtml).

#### Распределение баллов и сроки сдачи лабораторных работ

##### Семестр 6

№	Наименование лабораторной	Количество баллов
1.	Разработка физической модели базы данных с учетом декларативной ссылочной целостности	5
2.	Создание и модификация базы данных и таблиц	4
3.	Заполнение таблиц и модификация данных	5
4.	Разработка SQL запросов: виды соединений и шаблоны	7
5.	Разработка SQL запросов: запросы с подзапросами	7
6.	Хранимые процедуры	7
	Контрольная работа «Проектирование взаимодействия базы данных и приложения»	20
	Итого	55

##### Семестр 7

№	Наименование лабораторной	Количество баллов
7.	Объектно-реляционные базы данных. Проектирование и создание	14
8.	Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы	16
9.	Разработка документной базы данных	14
10.	Манипулирование данными в документной базе данных	16
	Контрольная работа «Триггеры. Обеспечение активной целостности данных базы данных»	20

	Итого	80
--	-------	----

При сдаче лабораторных работ позже заявленного срока (сессии) баллы за работуснижаются.

11.5. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся:

Структура пояснительной записки курсового проекта/ работы

Учебным планом не предусмотрено

Требования к оформлению пояснительной записки курсового проекта/ работы

Учебным планом не предусмотрено

11.6. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

Самостоятельная работа студента предполагает знакомство и использование источников информации, размещенных в сети Internet ведущими корпорациями – производителями современного программного обеспечения (таблица 9).

11.7. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости проходит в форме:

- контрольной работы «Проектирование взаимодействия базы данных и приложения» в 6 семестре
- контрольной работы «Триггеры. Обеспечение активной целостности базы данных» в 7 семестре.

Текущий контроль успеваемости в 6 семестре.

### Контрольная работа . «Проектирование взаимодействия базы данных и

#### приложения»Задание для выполнения в лабораторной работе

Создать приложение –форму с 2 способами подсоединения приложения к базе данных и:с двумя способами подсоединения базы данных:

1) Соединение через компоненты ADO.NET или его аналога с помощью строки связи. При этом отображаться должны 2 таблицы/элемента данных (не обязательно в форме таблицы, может быть просто текст): одна через DB Aware компонент (например, DataGridView в C#), с подключением источника данных, а другая через чтение результатов запроса SqlCommand в цикле.

2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных.

Вариант задания (предметная область) для базы данных используется сквозной единый для лабораторных работ 6 семестра (реляционной бд) и контрольных работ.

Контрольная работа выполняется в период заочного обучения и сдается в сессию, на лабораторных или практических занятиях с демонстрацией работоспособности кода и защитой.

Баллы, полученные за контрольную составляют 1/5 оценки за семестр(20 баллов).

В сессию должна быть завершена сдача всех контрольной и 6 лабораторных работ данного семестра.

Получение аттестационной оценки «отлично» при успешном прохождении промежуточной аттестации в форме экзамена возможно только при успешной сдаче всех лабораторных работ и контрольной.

В случае невыполнения и/или неуспешной сдачи 2 и более работ (лабораторных или контрольной работы) обучающийся, при успешном прохождении промежуточной аттестации в форме экзамена, не может получить аттестационную оценку выше "хорошо". В случае невыполнения и/или неуспешной сдачи 3 и более работ(лабораторных или контрольной работы), обучающийся, при успешном прохождении промежуточной аттестации в форме экзамена, не может получить аттестационную оценку выше "удовлетворительно".

Текущий контроль успеваемости в 7 семестре.

### Контрольная работа. «Триггеры. Обеспечение активной целостности данных базы данных»

#### Задание для выполнения в лабораторной работе

1)Реализовать для своей базы данных триггеры для всех событий (insert,delete, update) до и после.(6 триггеров) Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение из других предложенных , но не менее 2 различных

(- Вычисление/поддержание в актуальном состоянии вычисляемых (производных)атрибутов (полей);

- логирование (запись) изменений;
- обеспечения безопасности данных;
- проверка корректности проводимых действий.). Вычисляемые поля можно добавить при необходимости.

Вариант задания (предметная область) для базы данных используется сквозной единый для лабораторных работ 6 семестра (реляционной бд) и контрольных работ.

Контрольная работа выполняется в период заочного обучения и сдается в сессию, на лабораторных или практических занятиях с демонстрацией работоспособности кода и защитой.

Баллы, полученные за контрольные составляют 1/5 оценки за семестр (20 баллов).

В сессию должна быть завершена сдача всех контрольной и 4 лабораторных работ данного семестра.

В случае невыполнения и/или неуспешной сдачи 3 и более работ(контрольной и лабораторных), обучающийся, при успешном прохождении промежуточной аттестации в форме зачета, не может получить аттестационную оценку "зачтено".

В случае успешной сдачи баллов, лабораторные работы и контрольную превышающей 60 баллов возможно получение оценки "зачтено" за промежуточную аттестацию без дополнительных вопросов и задач, так как компетенции можно считать проверенными при защите лабораторных и контрольной работ.

11.8. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

- зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».

- В соответствии с учебным планом экзамен проходит в 6 семестре, зачет в 7.
- Вопросы и задачи для проведения экзамена представлены в таблице 15.
- Вопросы и задачи для проведения зачета представлены в таблице 16.
- Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программы высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».



Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой