

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

Кафедра № 43

УТВЕРЖДАЮ

Руководитель направления

ДОЦ., К.Т.Н., ДОЦ.

(должность, уч. степень, звание)

А.А. Ключарев

(инициалы, фамилия)

(подпись)

«15» июня 2021 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

Санкт-Петербург – 2021

Лист согласования рабочей программы дисциплины

Программу составил (а)

старший преподаватель  
(должность, уч. степень, звание)



11 июня 2021 г.  
(подпись, дата)

Н.В. Путилова  
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«15» июня 2021 г, протокол № 09-2020/21

Заведующий кафедрой № 43

д.т.н., проф.  
(уч. степень, звание)

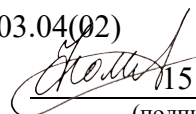


15 июня 2021 г.  
(подпись, дата)

М.Ю. Охтилев  
(инициалы, фамилия)

Ответственный за ОП ВО 09.03.04(02)

старший преподаватель  
(должность, уч. степень, звание)

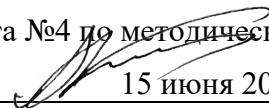


15 июня 2021 г.  
(подпись, дата)

А.А. Фоменкова  
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

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



15 июня 2021 г.  
(подпись, дата)

А.А. Ключарев  
(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-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, разработкой серверной части базы данных (в том числе триггеров и хранимых процедур), 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, консультации.

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

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часов.

Язык обучения по дисциплине «русский»

## 1. Перечень планируемых результатов обучения по дисциплине

### 1.1. Цели преподавания дисциплины

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-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»;
- «Структуры и алгоритмы обработки данных».

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

- «Разработка и анализ требований»
- «Экспертные системы»

## 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, ЗЕ/ (час)</b>	6/ 216	3/ 108	3/ 108
<b>Из них часов практической подготовки</b>			
<b>Аудиторные занятия, всего час.</b>	40	24	16
в том числе:			
лекции (Л), (час)	14	8	6
практические/семинарские занятия (ПЗ), (час)	4	4	
лабораторные работы (ЛР), (час)	22	12	10
курсовой проект (работа) (КП, КР), (час)			
экзамен, (час)	9	9	
<b>Самостоятельная работа, всего (час)</b>	167	75	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.	2	1	2		16
Тема 1.3 Нормализация и денормализация баз данных					
Тема 1.4 Роль баз данных в программных системах					
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	3	1	4		26
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	3	2	6		33
Тема 3.3.Использование представлений, управляющих конструкций в языке SQL					
Тема 3.4. Хранимые процедуры.					
Итого в семестре:	8	4	12		75
Семестр 7					
Раздел 3. Введение в язык SQL					
Тема 3.5. Триггеры, обеспечение активной целостности	1				18
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	2		5		35
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»	3		5		39
Тема 5.2. Документные базы данных					
Тема 5.3. Графовые базы данных					
Тема 5.4. Базы данных «Семейство столбцов»					
Итого в семестре:	6		10		92
Итого	14	4	22	0	167

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
<b>1</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 Тема 1.5. Установка и администрирование баз данных
<b>2</b>	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 конструкций 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
Семестр 6					
1	Проектирование базы данных: Критерии корректности определения связей и между сущностями нормализация реляционной базы данных	Кейс, занятие по моделированию реальных условий	1		1
2	Определение ключей отношений. Запросы на определение данных, манипулирование данными	Решение задач., кейс,	1		2
3	Построение сложных запросов на SQL: запросов с агрегатными функциями, с внешними соединениями, запросов с подзапросами	Решение задач.	1		3
4	Использование	Кейс, занятие по	1		3

	управляющих конструкций в языке SQL, хранимые процедуры	моделированию реальных условий			
Всего			4		

#### 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	2		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	2		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		22		

#### 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	129	55	74



дисциплины (ТО)			
Контрольные работы заочников (КРЗ)	14	6	8
Подготовка к промежуточной аттестации (ПА)	24	14	10
Всего:	167	75	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 экземпляро в в библиотеке (кроме электронн ых экземпляро в)
<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://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/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://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/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="http://citforum.ru/database/">http://citforum.ru/database/</a>	Базы данных
<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

#### 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
1	Open ModelSphere/ Draw.io/
2	MySQL Community Server
3	Microsoft Word или OpenOffice Writer/ LibreOffice Writer
4	Microsoft Visual Studio
5	MongoDB Community Server
6	PostgreSQL

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 таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	—
2	Вычислительная лаборатория кафедры 43	Б.М. 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; Экзаменационные билеты; Задачи; Тесты.
Зачет	Список вопросов; 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.

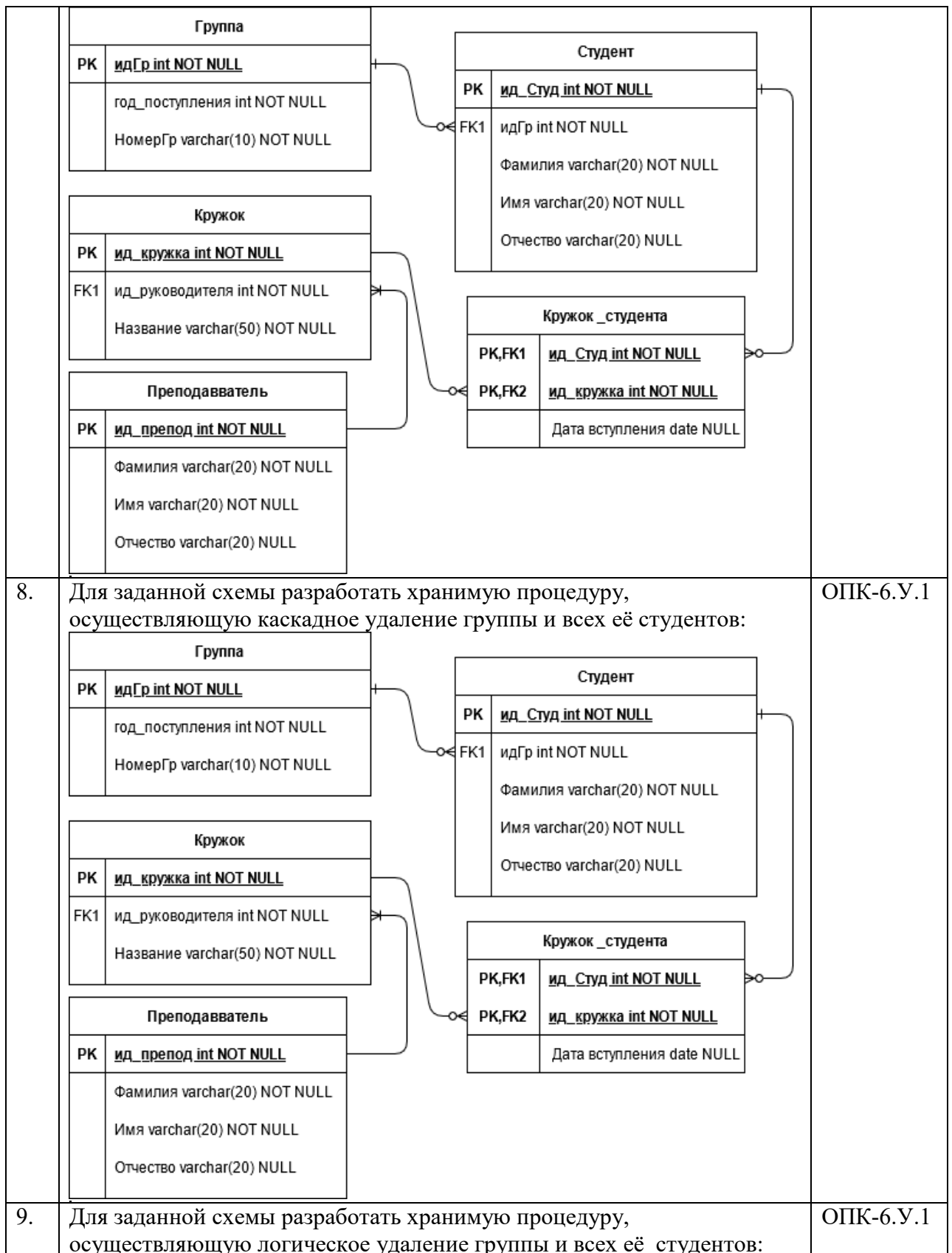
Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
		ОПК-6.3.1
		ОПК-6.У.1
1	История причины появления СУБД. Роль СУБД в программных системах	ОПК-6.3.1
2	История развития СУБД и модели данных.	ОПК-6.3.1
3	Объектно-ориентированные и NoSQL модели данных.	ОПК-6.3.1
4	Нормальные формы и денормализация	ОПК-6.3.1
5	Отношения и их свойства, ключи отношений	ОПК-6.3.1
6	Типы данных в языке SQL, оператор языка SQL для создания таблиц	ОПК-6.3.1
7	Операторы языка SQL для удаления и модификации таблиц	ОПК-6.3.1

8	Абстрактные операции манипулирования данными, их реализация в языке SQL	ОПК-6.3.1
9	Булевы операции над отношениями	ОПК-6.3.1
10	Операция выборки, ее свойства	ОПК-6.3.1
11	Операция проекции, ее свойства	ОПК-6.3.1
12	Операция соединения, ее свойства	ОПК-6.3.1
13	Операция деления	ОПК-6.3.1
14	Операция переименования атрибутов	ОПК-6.3.1
15	Операции эквисоединения, $\theta$ -выборки и $\theta$ -соединения	ОПК-6.3.1
16	Оператор выборки в языке SQL	ОПК-6.3.1
17	Агрегатные функции в операторе выборки языка SQL	ОПК-6.3.1
18	Объединение, пересечение, разность запросов в языке SQL	ОПК-6.3.1
19	Запросы с подзапросами в языке SQL	ОПК-6.3.1
20	Экзистенциальные запросы в языке SQL	ОПК-6.3.1
21	Виды соединений в языке SQL	ОПК-6.3.1
22	Трехзначная логика и обработка NULL-значений в языке SQL	ОПК-6.3.1
23	Представления в языке SQL	ОПК-6.3.1
24	Управляющие конструкции в языке SQL	ОПК-6.3.1
25	Хранимые процедуры в языке SQL	ОПК-6.3.1
	Задачи	
1.	Написать скрипт SQL, создающий таблицу из 2 столбцов, один из которых является строкой фиксированной длины и первичным ключом, а второй строкой ограниченной длины и внешним ключом	ОПК-6.У.1
2.	<p>Написать скрипт SQL, создающий базу данных по заданной схеме:</p>	ОПК-6.У.1
3.	Написать скрипт SQL, вставляющий данные в базу данных по заданной схеме:	ОПК-6.У.1

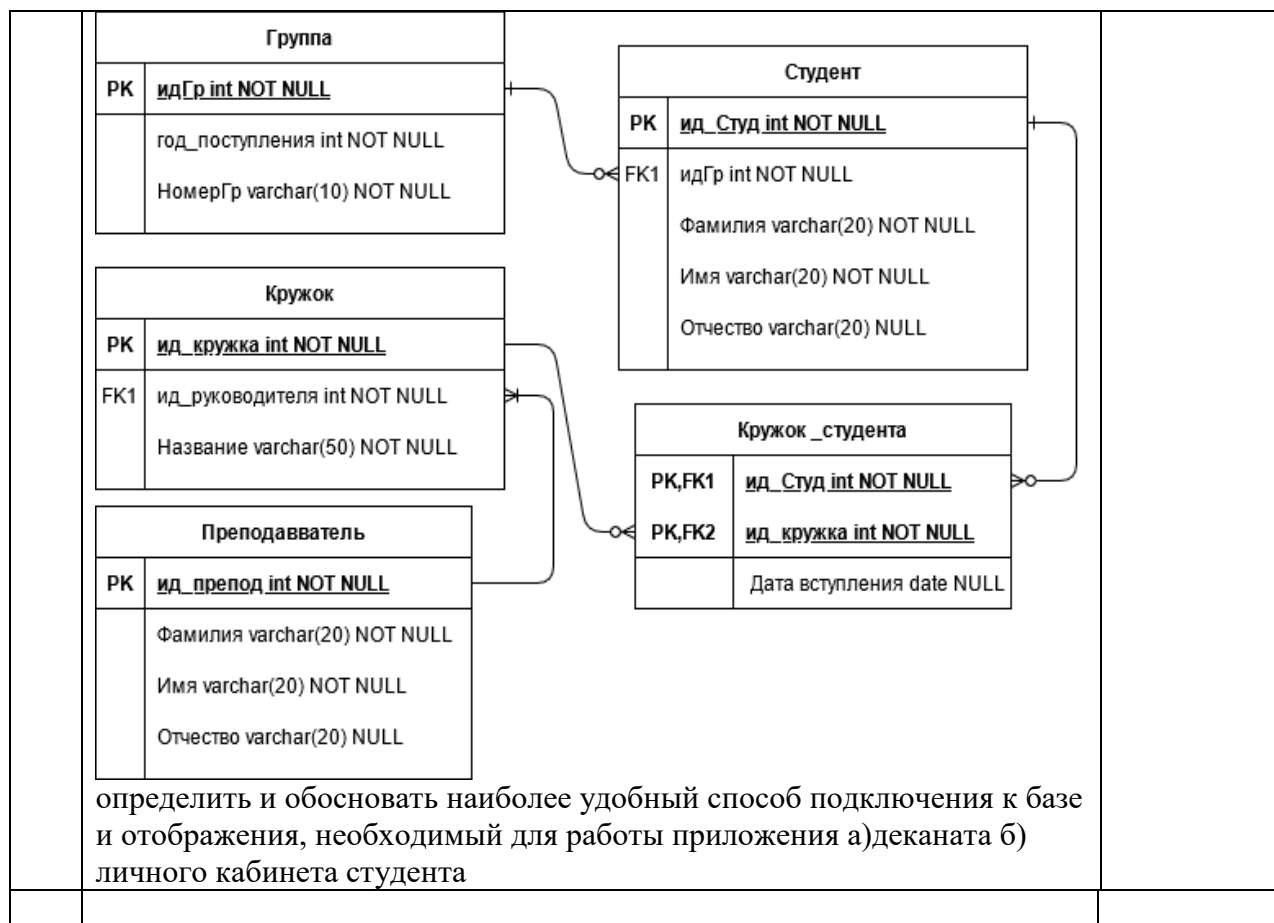
	<p><b>Группа</b></p> <table border="1"> <tr><td>PK</td><td><u>идГр int NOT NULL</u></td></tr> <tr><td></td><td>год_поступления int NOT NULL</td></tr> <tr><td></td><td>НомерГр varchar(10) NOT NULL</td></tr> </table> <p><b>Студент</b></p> <table border="1"> <tr><td>PK</td><td><u>ид_Студ int NOT NULL</u></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> <p><b>Кружок</b></p> <table border="1"> <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> <p><b>Преподаватель</b></p> <table border="1"> <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> <p><b>Кружок_студента</b></p> <table border="1"> <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>	PK	<u>идГр int NOT NULL</u>		год_поступления int NOT NULL		НомерГр varchar(10) NOT NULL	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	PK	<u>ид_кружка int NOT NULL</u>	FK1	ид_руководителя int NOT NULL		Название varchar(50) NOT NULL	PK	<u>ид_препод int NOT NULL</u>		Фамилия varchar(20) NOT NULL		Имя varchar(20) NOT NULL		Отчество varchar(20) NULL	PK,FK1	<u>ид_Студ int NOT NULL</u>	PK,FK2	<u>ид_кружка int NOT NULL</u>		Дата вступления date NULL	
PK	<u>идГр int NOT NULL</u>																																					
	год_поступления int NOT NULL																																					
	НомерГр varchar(10) NOT NULL																																					
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																																					
PK	<u>ид_кружка int NOT NULL</u>																																					
FK1	ид_руководителя int NOT NULL																																					
	Название varchar(50) NOT NULL																																					
PK	<u>ид_препод int NOT NULL</u>																																					
	Фамилия varchar(20) NOT NULL																																					
	Имя varchar(20) NOT NULL																																					
	Отчество varchar(20) NULL																																					
PK,FK1	<u>ид_Студ int NOT NULL</u>																																					
PK,FK2	<u>ид_кружка int NOT NULL</u>																																					
	Дата вступления date NULL																																					
4.	<p>Написать скрипт SQL, удаляющий кружки, в которых нет студентов данные из базы данных по заданной схеме:</p>	ОПК-6.У.1																																				
5.	<p>Написать запрос SQL, выбирающий кружок с количеством студентов больше среднего по заданной схеме:</p>	ОПК-6.У.1																																				

	<p>The ER diagram shows four tables: <b>Группа</b>, <b>Студент</b>, <b>Кружок</b>, and <b>Кружок_студента</b>. <b>Группа</b> has a primary key <u>идГр int NOT NULL</u> and attributes <code>год_поступления int NOT NULL</code> and <code>НомерГр varchar(10) NOT NULL</code>. <b>Студент</b> has a primary key <u>ид_Студ int NOT NULL</u> and attributes <code>идГр int NOT NULL</code> (FK1), <code>Фамилия varchar(20) NOT NULL</code>, <code>Имя varchar(20) NOT NULL</code>, and <code>Отчество varchar(20) NULL</code>. <b>Кружок</b> has a primary key <u>ид_кружка int NOT NULL</u> and attributes <code>ид_руководителя int NOT NULL</code> (FK1) and <code>Название varchar(50) NOT NULL</code>. <b>Кружок_студента</b> has a composite primary key <u>ид_Студ int NOT NULL</u> (PK,FK1) and <u>ид_кружка int NOT NULL</u> (PK,FK2), and an attribute <code>Дата вступления date NULL</code>. Relationships: <b>Группа</b> (1) to <b>Студент</b> (many); <b>Кружок</b> (1) to <b>Кружок_студента</b> (many); <b>Студент</b> (1) to <b>Кружок_студента</b> (many).</p>	
6.	<p>Написать запрос SQL, выбирающий кружок , в который ходят все студенты группы 4136 по заданной схеме:</p>	ОПК-6.У.1
7.	<p>Написать запрос SQL (в 3 вариантах), выбирающий кружок , в который не ходят студенты группы 4136 по заданной схеме:</p>	ОПК-6.У.1



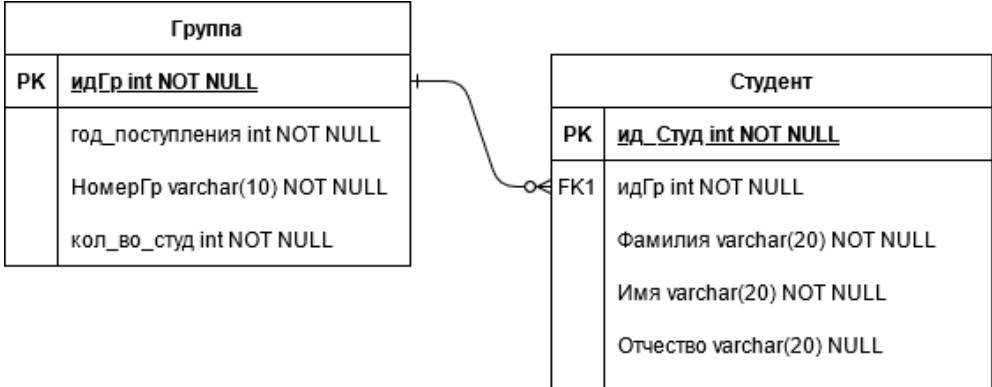
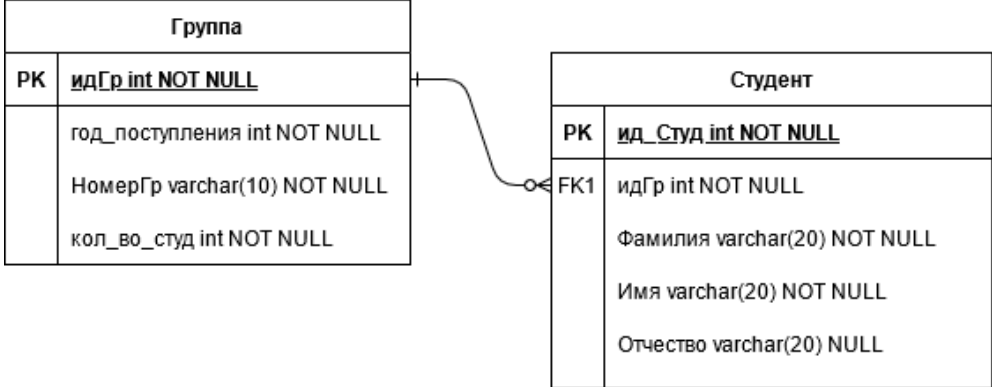


10.	<p>Для заданной схемы</p> <p>написать скрипт на SQL , изменяющий первичный ключ НомерГр, таблицы Группа на составной первичный ключ НомерГр,Год_Поступления и все внешние ключи, затронутые изменением, соответственно.</p>	ОПК-6.У.1
11.	<p>Для заданной схемы</p> <p>написать скрипт на SQL , добавляющий в таблицу Группа столбец количество_студентов, и заполняющий его корректными данными.</p>	ОПК-6.У.1
12.	<p>Для базы данных по заданной схеме</p>	ОПК-6.У.1



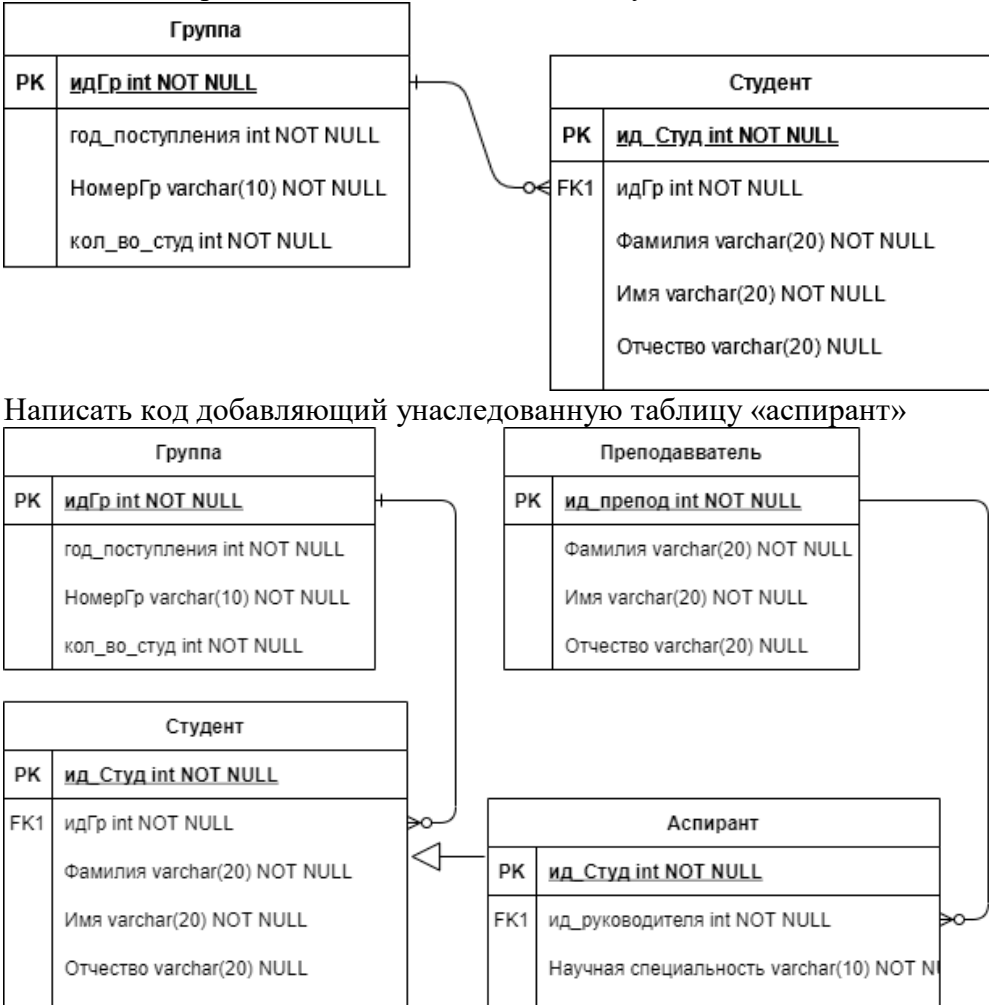
Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.  
Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1	Триггеры в языке SQL	ОПК-6.3.1
2	Индексация данных	ОПК-6.3.1
3	Защита информации в базах данных	ОПК-6.3.1
4	Целостность баз данных и параллельный доступ	ОПК-6.3.1
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-

		6.3.1
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> 	ОПК-6.У.1
2.	<p>Для базы данных по заданной схеме написать триггер, осуществляющие удаление группы, если в ней не осталось студентов.</p> 	ОПК-6.У.1
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 заданной схеме	ОПК-6.У.1

	<table border="1"> <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> </table> <table border="1"> <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 border="1"> <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>	Группа		PK	<u>идГр int NOT NULL</u>		год_ поступления int NOT NULL		НомерГр varchar(10) 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	<table border="1"> <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> <table border="1"> <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>	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_студента		PK,FK1	<u>ид_Студ int NOT NULL</u>	PK,FK2	<u>ид_кружка int NOT NULL</u>		Дата вступления date NULL	<p>определить и обосновать уровень изоляции транзакций, необходимый для работы приложения а) деканата б) личного кабинета студента</p>
Группа																																																	
PK	<u>идГр int NOT NULL</u>																																																
	год_ поступления int NOT NULL																																																
	НомерГр varchar(10) 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	идГр int NOT 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																																																
5.	Для предметной области «учебный план» (дисциплины, кафедры, знания и умения полученные в результате освоения дисциплин) предложить и обосновать выбор модели данных СУБД. Предложить 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. Предложить 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. Предложить примерную схему базы данных.	ОПК-6.У.1																																															
8.	Для реляционной базы со схемой	ОПК-6.У.1																																															

	<p>ER diagram showing relationships between Group, Student, Club, and Teacher tables. Group is the primary key for Student (FK1). Club has a primary key and a foreign key to Group. Club has a primary key and a foreign key to Student (FK1). Club has a primary key and a foreign key to Student (FK1) and Club (FK2).</p>	
	создать соответствующую ей (по данным) документную базу данных	
9.	<p>Для реляционной базы со схемой</p> <p>ER diagram showing relationships between Group, Student, Club, and Teacher tables. Group is the primary key for Student (FK1). Club has a primary key and a foreign key to Group. Club has a primary key and a foreign key to Student (FK1). Club has a primary key and a foreign key to Student (FK1) and Club (FK2).</p>	ОПК-6.У.1
10.	<p>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.</p>	ОПК-6.У.1
11.	<p>Для документной базы данных следующей схемы (коллекция студентов,</p>	ОПК-

	<p>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</p>	6.У.1
12.	<p>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»</p>	ОПК-6.У.1
13.	<p>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, прикрепленные в позапрошлом году</p>	ОПК-6.У.1
14.	<p>Для объектно-реляционной базы данных следующей схемы</p>  <p>Написать код добавляющий унаследованную таблицу «аспирант»</p>	ОПК-6.У.1
15.	<p>Для объектно-реляционной базы данных следующей схемы</p>	ОПК-6.У.1

	<p>определить другие примеры наследования</p>	
16.	<p>Для объектно-реляционной базы данных следующей схемы</p> <p>предложить пользовательский оператор и реализовать его для postgres</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>Назовите описываемую модель данных одним словом.          Модель данных, основанная на графах, где в узле графа стоит произвольная структура и ребра графа (связь между узлами) могут легко формироваться во время работы с базой данных в зависимости от текущих взаимосвязей между объектами-узлами графа.</p> <p><b>Графовая</b></p>	ОПК-6.3.1
2.	<p>Назовите описываемую модель данных одним словом.          Модель данных, реализующая структуру данных «дерево» с узлом, содержащим изначально жестко заданную табличную структуру.</p> <p><b>Иерархическая</b></p>	ОПК-6.3.1
3.	<p>Назовите описываемую модель данных одним словом.          Модель данных, основанная на графах, где для реализации связей в предметной области жестко задана связь между узлами графа(ребра), и в узле графа стоит табличная структура изначально жестко заданной структуры.</p> <p><b>Сетевая</b></p>	ОПК-6.3.1
4.	<p>Назовите описываемую модель данных одним словом.          Модель данных , где основные единицы хранения представляют собой самоописываемые гибкие иерархические древовидные структуры данных, которые хранятся как пара объектов типа "ключ-значение", в которых значение объекта допускает проверку, за счёт прозрачности.</p> <p><b>Документная</b></p>	ОПК-6.3.1
5.	<p>Продолжите фразу:          Домен атрибута — это</p> <ul style="list-style-type: none"> <li>а. диапазон допустимых значений атрибута</li> <li>б. -тип данных атрибута</li> <li>в. множество значений атрибута, находящихся в данный момент в теле отношения</li> <li>г. множество атрибутов, объединенных одним типом данных</li> </ul>	ОПК-6.3.1
6.	<p>Продолжите фразу:          В отношении одинаковые строки</p> <ul style="list-style-type: none"> <li>а. могут существовать</li> <li>б. <b>не могут существовать</b></li> <li>в. могут возникнуть как результаты операций реляционной алгебры</li> <li>г. могут существовать в ненормализованной базе данных</li> </ul>	ОПК-6.3.1
7.	<p>Какой ключ описывается такими словами (ответьте одним словом)?          Потенциальный ключ, выбранный для уникальной идентификации всех остальных значения атрибутов в любой строке.</p> <p><b>Первичный</b></p>	ОПК-6.3.1
8.	<p>«Отношение без зависимостей соединения.»          Это требование какой нормальной формы?</p> <p><b>Пятая нормальная форма</b></p>	ОПК-6.3.1
9.	<p>«Каждый детерминант отношения является потенциальным ключом.»</p>	ОПК-



	<p>Это требование какой нормальной формы?  <b>Нормальная форма Бойса –Кодда</b></p>	6.3.1																				
10.	<p>Какой по архитектуре тип баз данных не требует отдельного серверного процесса или системы для работы и эффективно скрывает или минимизирует взаимодействие с базой данных конечными пользователями приложения?</p> <p>а. Персональные  б. Клиентские  <b>в. Встроенные</b>  г. Резидентные</p>	ОПК-6.3.1																				
11.	<p>Дана предметная область</p> <p><b>R1</b></p> <table border="1"> <thead> <tr> <th>Студент</th> <th>дисциплина</th> <th>тема курсового</th> <th>оценка</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p><b>R2</b></p> <table border="1"> <thead> <tr> <th>группа</th> <th>дисциплина</th> <th>вид занятия</th> <th>семестр</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p><b>R3</b></p> <table border="1"> <thead> <tr> <th>Студент</th> <th>группа</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> <p>Выражение:</p> $\Pi_{\{\text{Студент, семестр}\}} \left( \sigma_{\text{вид занятия} = \text{"лабораторные"}} (R2 \bowtie R3) \right) \div \Pi_{\{\text{семестр}\}} (R2)$ <p>Что находит данное выражение?</p> <p>а. Студенты, у которых есть лабораторные занятия во всех семестрах  б. Семестры, в которых у всех студентов есть лабораторные занятия  в. Все студенты, у которых есть лабораторные занятия в семестре  г. Все семестры, в которых у студентов есть лабораторные занятия</p>	Студент	дисциплина	тема курсового	оценка					группа	дисциплина	вид занятия	семестр					Студент	группа			ОПК-6.3.1
Студент	дисциплина	тема курсового	оценка																			
группа	дисциплина	вид занятия	семестр																			
Студент	группа																					
12.	<p>Как называется вид подзапроса, в котором есть ссылка на некоторую «внешнюю» по отношению к подзапросу таблицу?  <b>Коррелированный</b></p>	ОПК-6.3.1																				
13.	<p>Дана таблица студентов (student)с двумя логическими полями (возможность претендовать на красный диплом is_red_diploma и получал ли гранты is_special_grant )</p>	ОПК-6.3.1																				

id_s	surname	name	is_red_diploma	is_special_grant
1	Иванов	Иван	true	null
2	Петров	Петр	null	false
3	Сидоров	Сидор	false	true

Какой результат выдаст запрос ниже?

SELECT is\_red\_diploma AND is\_special\_grant FROM student WHERE

id\_s=1

**Null**

14.	<p>Как называется подход к формированию представления, когда при обращении к представлению осуществляется подстановка вместо представления формирующих его операторов SQL, который обращается к базовым таблицам?</p> <p><b>Замена представления</b></p>	ОПК-6.3.1
15.	<p>К таким переменным можно получить доступ без объявления, им может быть присвоено значение из ограниченного набора типов данных они зависят от сеанса, имея при этом сквозную видимость.</p> <p><b>а. Пользовательские переменные</b>          б. Сеансовые переменные          в. Локальные переменные          г. Системные переменные</p>	ОПК-6.3.1
16.	<p>Какой тип данных является числовым типом с фиксированной запятой?</p> <p><b>Numeric</b>  <b>Decimal</b></p>	ОПК-6.3.1
17.	<p>Какие уровни изоляции транзакций защищают от фантомных (строк)?</p> <p><b>а. SERIALIZABLE</b>          б. READ UNCOMMITTED          в. READ COMMITTED          г. REPEATABLE READ</p>	ОПК-6.3.1
18.	<p>Как расшифровывается термин В*-дерево, являющееся одним из типов индексов по организации хранения ?</p> <p><b>а. Сбалансированное дерево</b>          б. Бинарное дерево          в. Двухуровневое дерево          г. Сильноветвящееся дерево</p>	ОПК-6.3.1
19.	<p>Дан триггер</p> <pre> delimiter // create trigger ins_dis before insert on discipline for each row if not exists(select * from language where id_lang= new.id_lang) then insert into language (id_lang,lang_name) values (new.id_lang,new.l end if; end// delimiter ; </pre>	ОПК-6.3.1

	<p>Сколько раз выполнится триггер при выполнении следующей команды?</p> <pre>insert into discipline (id_dis,name,id_lang) values (1,'ООП',1), (2,'Информатика',1), (3,'Программирование',1);</pre> <p>а. 3 б. 1 в. Неизвестно, так как неизвестна корректность вставляемых данных</p>																									
20.	<p>Выберите все корректные варианты из предложенных</p> <p>В операторе создания таблиц можно</p> <p>а. указать способы поддержания ссылочной целостности б. указать внешний ключ таблицы в. указать значение по умолчанию для столбца г. указать вторичный ключ в таблицы д. создать домен столбца е. назначить псевдоним по умолчанию для таблицы</p>	ОПК-6.У.1																								
21.	<p>Какой или какие из операторов корректно добавляют столбец к заполненной таблице</p> <table border="1" data-bbox="339 958 721 1142"> <thead> <tr> <th colspan="4">student</th> </tr> </thead> <tbody> <tr> <td>id_st</td> <td>int</td> <td>&lt;pk&gt;</td> <td>not null</td> </tr> <tr> <td>id_group</td> <td>int</td> <td>&lt;fk&gt;</td> <td>null</td> </tr> <tr> <td>surname</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> <tr> <td>name</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> <tr> <td>patronym</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> </tbody> </table> <p>а. ALTER TABLE student ADD email varchar(15) null; б. ALTER TABLE student ADD COLUMN email varchar(15) null; в. ALTER TABLE student ADD email varchar(15) not null; г. ADD TO student email varchar(15) null;</p>	student				id_st	int	<pk>	not null	id_group	int	<fk>	null	surname	varchar(20)		null	name	varchar(20)		null	patronym	varchar(20)		null	ОПК-6.У.1
student																										
id_st	int	<pk>	not null																							
id_group	int	<fk>	null																							
surname	varchar(20)		null																							
name	varchar(20)		null																							
patronym	varchar(20)		null																							
22.	<p>Какой или какие из операторов полностью корректно вставляет данные в изображенную ниже таблицу с учетом того, что ключ не автоинкрементный?</p> <table border="1" data-bbox="339 1473 721 1657"> <thead> <tr> <th colspan="4">student</th> </tr> </thead> <tbody> <tr> <td>id_st</td> <td>int</td> <td>&lt;pk&gt;</td> <td>not null</td> </tr> <tr> <td>id_group</td> <td>int</td> <td>&lt;fk&gt;</td> <td>null</td> </tr> <tr> <td>surname</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> <tr> <td>name</td> <td>varchar(20)</td> <td></td> <td>not null</td> </tr> <tr> <td>patronym</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> </tbody> </table> <p>а. INSERT INTO student (id_st, id_group, surname, name, patronym) VALUES (1, 1,'Иванов','Иван', 'Иванович'); б. INSERT INTO student VALUES (1, 1,'Иванов','Иван', 'Иванович'); в. INSERT INTO student (id_st, surname) VALUES (1, 'Иванов'); г. INSERT INTO student (1, 1,'Иванов','Иван', 'Иванович');</p>	student				id_st	int	<pk>	not null	id_group	int	<fk>	null	surname	varchar(20)		null	name	varchar(20)		not null	patronym	varchar(20)		null	ОПК-6.У.1
student																										
id_st	int	<pk>	not null																							
id_group	int	<fk>	null																							
surname	varchar(20)		null																							
name	varchar(20)		not null																							
patronym	varchar(20)		null																							
23.	<p>Дана таблица преподавателей (Teacher) и обновленная таблица преподавателей (new_Teacher) в базе данных расписания вуза.</p>	ОПК-6.У.1																								

Teacher			
id_T	int	<pk>	not null
surname	varchar(25)		not null
name	varchar(25)		null
patronym	varchar(25)		null
degree	varchar(30)		null

new_Teacher			
id_T	int	<pk>	not null
surname	varchar(25)		not null
name	varchar(25)		null
patronym	varchar(25)		null
degree	varchar(30)		null

Какой или какие из операторов полностью корректно приводит данные таблицы Teacher в соответствии с данными таблицы new\_Teacher?

а.

```
MERGE INTO Teacher USING new_Teacher
ON Teacher.id_T= new_Teacher.id_T
WHEN MATCHED THEN
UPDATE SET Teacher.surname= new_Teacher.surname,
Teacher.name= new_Teacher.name, Teacher.patronym=
new_Teacher.patronym, Teacher.degree= new_Teacher.degree
WHEN NOT MATCHED BY SOURCE THEN
DELETE
WHEN NOT MATCHED THEN
INSERT(id_T, surname, name, patronym,degree)
VALUES (new_Teacher.id_T, new_Teacher.surname,
new_Teacher.name, new_Teacher.patronym, new_Teacher.degree)
```

б.

```
MERGE INTO Teacher USING new_Teacher
ON Teacher.id_T= new_Teacher.id_T
WHEN MATCHED THEN
UPDATE
WHEN NOT MATCHED BY SOURCE THEN
DELETE
WHEN NOT MATCHED THEN
INSERT
```

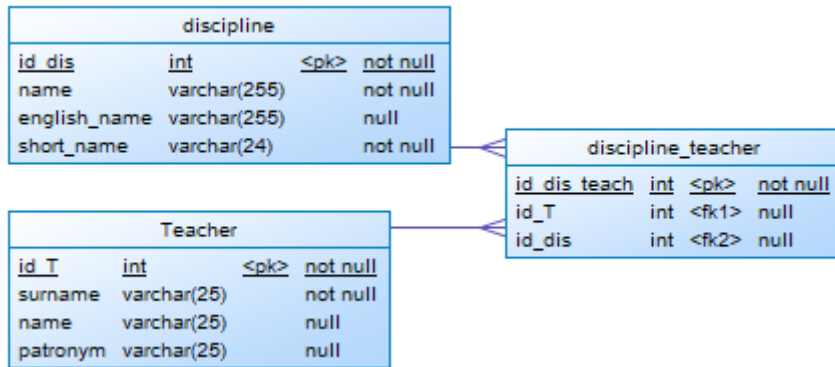
в.

```
MERGE INTO Teacher USING new_Teacher
ON Teacher.id_T= new_Teacher.id_T
WHEN MATCHED THEN
UPDATE SET Teacher.surname= new_Teacher.surname, Teacher.name=
new_Teacher.name, Teacher.patronym= new_Teacher.patronym,
Teacher.degree= new_Teacher.degree
WHEN NOT MATCHED THEN
INSERT(id_T, surname, name, patronym, degree)
VALUES (new_Teacher.id_T, new_Teacher.surname, new_Teacher.name,
new_Teacher.patronym, new_Teacher.degree)
```

г.

```
MERGE INTO Teacher USING new_Teacher
ON Teacher.id_T= new_Teacher.id_T
WHEN MATCHED THEN
UPDATE SET Teacher.surname= new_Teacher.surname, Teacher.name=
new_Teacher.name, Teacher.patronym= new_Teacher.patronym,
Teacher.degree= new_Teacher.degree
```

	<p>WHEN NOT MATCHED BY SOURCE THEN  DELETE FROM Teacher  WHEN NOT MATCHED THEN  INSERT(id_T, surname, name, patronym, degree)  VALUES (new_Teacher.id_T, new_Teacher.surname, new_Teacher.name,  new_Teacher.patronym, new_Teacher.degree)</p>																									
24.	<p>Какой или какие из операторов полностью корректно удалят из  изображенной ниже таблицы студентов с идентификатором большим  тысячи?</p> <table border="1"> <thead> <tr> <th colspan="4">student</th> </tr> </thead> <tbody> <tr> <td>id_st</td> <td>int</td> <td>&lt;pk&gt;</td> <td>not null</td> </tr> <tr> <td>id_group</td> <td>int</td> <td>&lt;fk&gt;</td> <td>null</td> </tr> <tr> <td>sumame</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> <tr> <td>name</td> <td>varchar(20)</td> <td></td> <td>not null</td> </tr> <tr> <td>patronym</td> <td>varchar(20)</td> <td></td> <td>null</td> </tr> </tbody> </table> <p>а. <b>DELETE FROM student WHERE id_st&gt;1000;</b>  б. <b>DELETE FROM student WHERE not(id_st&lt;=1000);</b>  в. DELETE student WHERE id_st&gt;1000;  г. TRUNCATE TABLE student WHERE id_st&gt;1000;</p>	student				id_st	int	<pk>	not null	id_group	int	<fk>	null	sumame	varchar(20)		null	name	varchar(20)		not null	patronym	varchar(20)		null	ОПК- 6.У.1
student																										
id_st	int	<pk>	not null																							
id_group	int	<fk>	null																							
sumame	varchar(20)		null																							
name	varchar(20)		not null																							
patronym	varchar(20)		null																							
25.	<p>Существует таблица групп,</p> <table border="1"> <thead> <tr> <th colspan="4">st_group</th> </tr> </thead> <tbody> <tr> <td>id_gr</td> <td>int</td> <td>&lt;pk&gt;</td> <td>not null</td> </tr> <tr> <td>num_gr</td> <td>varchar(10)</td> <td></td> <td>not null</td> </tr> </tbody> </table> <p>где номер каждой группы имеет следующую структуру: ABCDX  , где А-обозначение факультета/института (М для ФПТИ, Z- для заочного  отделения 10 института и цифра института для номерных),  В- значащая цифра года поступления (7 для 2017,2007,2027 годов  поступления),  С- номер кафедры на факультете (т.е. полный номер кафедры это АС),  D-номер группы на выпускающей кафедре (1 символ),  Х- необязательный Набор признаков группы, которые могут указывать на  контрактное обучение (К), ускоренное обучение (с), заочное обучение на  факультете (з) (не заочном) и т.д.  Какой шаблон (введите без кавычек) необходимо вставить в запрос ниже,  чтобы получить все группы 2018 года поступления ( Считаем, что в  текущий момент в базе находятся только актуальные группы, т.е группы  других десятилетий мешать не будут.)  Select * from st_group where num_gr like 'шаблон'  _8%  _8_%  _8__%</p>	st_group				id_gr	int	<pk>	not null	num_gr	varchar(10)		not null	ОПК- 6.У.1												
st_group																										
id_gr	int	<pk>	not null																							
num_gr	varchar(10)		not null																							
26.	<p>Дана предметная область , описывающая ведение дисциплин  (discipline) преподавателями (teacher).</p>	ОПК- 6.У.1																								



Какой или какие из запросов вернёт преподавателей, не преподающих ни одной дисциплины?

- a. `SELECT Teacher.* FROM Teacher LEFT JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T is null;`
- б. `SELECT Teacher.* FROM Teacher RIGHT JOIN discipline_teacher ON discipline_teacher.id_T = Teacher.id_T WHERE discipline_teacher.id_T is null;`
- в. `SELECT Teacher.* FROM Teacher JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T is null;`
- г. `SELECT Teacher.* FROM Teacher LEFT JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T= null;`

27.

Дана сводная таблица по работам студентов

Работы

Студент	Дисциплина	Работа	Балл
Иванов	ООП	1	8
Петров	ООП	2	6
Сидоров	ПБД	1	4
Иванов	ПБД	1	Null

Какое число вернет запрос ниже?

`SELECT avg(Балл) from Работы`

- a. 6
- б. 4,5
- в. Null
- г. Ничего не выведет, выдаст ошибку, т.к. агрегатные функции не работают с null-значениями

ОПК-6.Y.1

28.

Даны таблицы

daily\_temp

city	day	temperature
Москва	14.05.2020	15
Симферополь	15.05.2020	23
Мурманск	16.05.2019	5

ОПК-6.Y.1

daily\_temp\_new

city	day	temperature
Москва	14.05.2020	15
Симферополь	14.05.2020	23
Санкт-Петербург	15.05.2020	15

SELECT \* FROM daily\_temp

ОПЕРАТОР

SELECT \* FROM daily\_temp\_new

Какой оператор необходимо подставить вместо слова оператор, чтобы получить результат ниже?

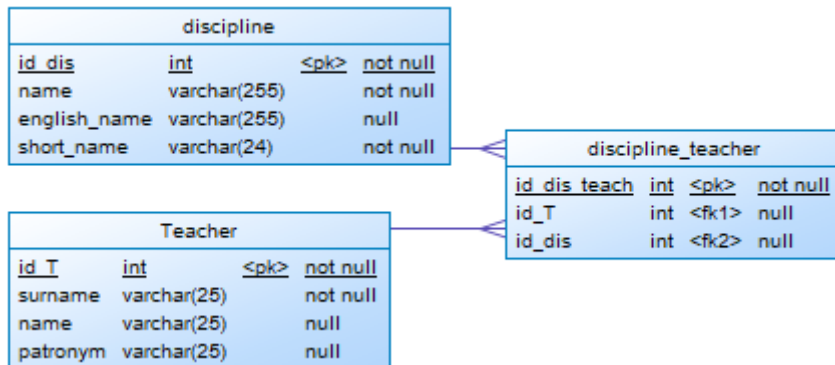
city	day	temperature
Москва	14.05.2020	15

INTERSECT

29.

Дана предметная область , описывающая ведение дисциплин (discipline) преподавателями (teacher).

ОПК-6.У.1



Какой или какие из запросов вернёт преподавателей ведущих все дисциплины со словом «технологии» в названии?

- SELECT distinct Teacher.\* FROM teacher WHERE NOT EXISTS (SELECT \* FROM discipline WHERE discipline.name LIKE '%технологии%' AND NOT EXISTS (SELECT \* FROM discipline\_teacher as d\_t WHERE d\_t.id\_T=Teacher.id\_T AND d\_t.id\_dis=discipline.id\_dis));**
- SELECT distinct Teacher.\* FROM Teacher WHERE NOT EXISTS (SELECT \* FROM discipline WHERE NOT EXISTS (SELECT \* FROM discipline\_teacher as d\_t JOIN discipline as d on d\_t.id\_dis=d.id\_dis WHERE d\_t.id\_T=Teacher.id\_T AND d\_t.id\_dis=discipline.id\_dis AND d.name LIKE '%технологии%' ));
- SELECT distinct Teacher.\* FROM teacher JOIN discipline\_teacher

	<pre> as d_t ON d_t.id_T=Teacher.id_T WHERE EXISTS (SELECT * FROM discipline WHERE discipline.name LIKE ‘%технологии%’ AND d_t.id_dis=discipline.id_dis); г. SELECT distinct Teacher.* FROM teacher JOIN discipline_teacher as d_t ON d_t.id_T=Teacher.id_T JOIN discipline as d on d_t.id_dis=d.id_dis WHERE discipline.name LIKE ‘%технологии%’ AND NOT EXISTS (SELECT * from Teacher as T WHERE NOT EXISTS (SELECT * FROM discipline_teacher as d_t1 WHERE d_t1.id_T=T.id_T AND d_t1.id_dis=discipline.id_dis)); </pre>	
30.	<p>Дана таблица студент</p> <pre> create table student ( id_st          int auto_increment primary key, id_group       int, surname        varchar(20), name           varchar(20) not null, patronym       varchar(20), ); </pre> <p>Код которой/которых из процедур правильно добавляет в неё данные и возвращает идентификатор добавленного значения?</p> <pre> delimiter // CREATE PROCEDURE ins_student (id_gr_ int,name_ varchar(15), surname_ varchar(20),patronym_ varchar(25), out id_st_new int) BEGIN insert into student ( surname, name,patronym,id_gr) VALUES (surname_,name_,patronym_,id_gr_);  set id_st_new=LAST_INSERT_ID(); END;// delimiter ; </pre> <p>a. <span style="float: right;">+</span></p>	ОПК-6.Y.1



```

delimiter //
CREATE PROCEDURE ins_student (id_gr_ int,name_ varchar(15),
surname_ varchar(20),patronym_ varchar(25), out id_st_new int)
BEGIN
set id_st_new=LAST_INSERT_ID();
insert into student (id_st, surname, name,patronym,id_gr)
VALUES (id_st_new,surname_,name_,patronym_,id_gr_);

END;//

```

б. delimiter ;

```

delimiter //
CREATE PROCEDURE ins_student (id_gr_ int,name_ varchar(15),
surname_ varchar(20),patronym_ varchar(25), id_st_new int)
BEGIN
insert into student ( surname, name,patronym,id_gr)
VALUES (surname_,name_,patronym_,id_gr_);

set id_st_new=LAST_INSERT_ID();
END;//

```

в. delimiter ;

```

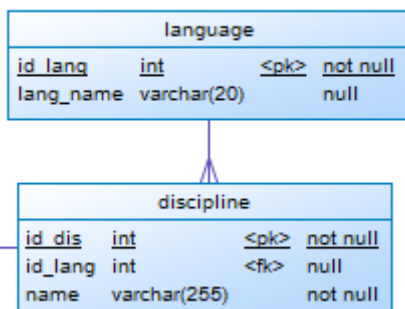
CREATE PROCEDURE ins_student (id_gr_ int,name_ varchar(15),
surname_ varchar(20),patronym_ varchar(25), out id_st_new int)
BEGIN
set id_st_new=( select ifnull(max(id_st)+1,0) from student);
insert into student (id_st, surname, name,patronym,id_gr)
VALUES (id_st_new,surname_,name_,patronym_,id_gr_);
END;//

```

г. delimiter ;

31.

Даны таблица языков преподавания и дисциплин



На таблицу дисциплин написан триггер before insert

ОПК-  
6.У.1

	<pre> create trigger ins_dis before insert on discipline for each row if not exists(select * from language where id_lang= new.id_lang)   then signal sqlstate '45000' set message_text = 'Нет такого языка'; end if; end// delimiter ; </pre> <p>Как передать в триггер параметры, чтобы добавить отсутствующий язык ?</p> <p>а. <b>Никак</b></p> <pre> delimiter // create trigger ins_dis (new_lang varchar (20)) before insert on discipline for each row if not exists(select * from language where id_lang= new.id_lang)   then insert into language (id_lang,lang_name) values (new.id_lang,   end if; end// </pre> <p>б. delimiter ;</p> <p>в. Не надо передавать, они уже есть в new</p>	
--	--	--

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

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
<b>1</b>	<b>Проектирование взаимодействия базы данных и приложения (6 семестр)</b>
1.1	<p>Создать для базы данных программы для рисования графов 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. 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>
1.2	<p>Создать для базы данных Садоводства 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</p>

	<p>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 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>
1.3	<p>Создать для базы данных ТСЖ 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. 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</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. 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</p>

	<p>третьей таблицы /элемента данных 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. 2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных 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. 2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> <li>а. виды спорта, на которые есть безлимитные абонементы (со словом безлимит 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. 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. 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</p>

	<p>третьей таблицы /элемента данных 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. 2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных 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. 2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных 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. 2) С использованием технологии ORM (Object-Relational Mapping) с отображение третьей таблицы /элемента данных База данных должна иметь структуру позволяющую реализовать следующие</p>

	<p>запросы:</p> <ul style="list-style-type: none"> <li>а. экскурсии в музей (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 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 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 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> <ol 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> </ol>
<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> <ol 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> </ol>
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> <ol 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> </ol>
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> <ol style="list-style-type: none"> <li>а. квартиры владельцев, отчества которых заканчиваются на 'ич'</li> </ol>

	<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> <ul style="list-style-type: none"> <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>
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> <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>
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> <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, остановившийся 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>
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> <ul style="list-style-type: none"> <li>з. виды спорта, на которые есть безлимитные абонементы (со словом безлимит в названии типа)</li> <li>и. вид спорта, по которому занятия ведут различные тренеры</li> <li>к. вид спорта, по которому не куплено ни одного абонемента</li> <li>л. тренеры, ведущие занятия по виду спорта с минимальным число занятий в абонементе</li> <li>м. тренеры, которые ведут занятия по максимальному количеству видов спорта</li> <li>н. тренеры, не ведущие фитнес</li> <li>о. абонемент, на все виды спорта, связанные с единоборствами, заканчивающимися на «до»</li> </ul>
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. 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>
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> <ol 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> </ol>
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> <ol 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> </ol>
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> <ol 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> </ol>
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> <ol 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> </ol>
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 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>
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. Методические указания для обучающихся по освоению дисциплины

#### **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;
- приобретение навыков методической обработки материалов (выделение главных мыслей и положений, формулировка конкретных выводов);
- получение точного понимания всех необходимых терминов и понятий.

Лекции обеспечивают теоретическое изучение дисциплины «Проектирование баз данных». На лекциях излагаются теоретические основы баз данных, методология их создания, а так же начальные сведения и указание источников для изучения основных возможностей языка SQL.

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 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.

### 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).

## Перечень заданий на лабораторные работы

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 операторов 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 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

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

Распределение баллов и сроки сдачи лабораторных работ

Семестр 6

№	Наименование лабораторной	Количество во баллов
1.	Разработка физической модели базы данных с учетом декларативной ссылочной целостности	5
2.	Создание и модификация базы данных и таблиц базы данных	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 во 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 работу снижаются.

Подробные методические указания по прохождению лабораторных работ, структуре и оформлению отчета приведены в Методических указаниях по выполнению лабораторных работ, расположенных на сервере кафедры 43, доступном из лабораторий кафедры по адресу:  
 z:/Методическое обеспечение каф 43/Проектирование баз данных/МУЛР ПБД 2 сем.pdf

**11.3. Методические указания для обучающихся по прохождению практических занятий**

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.

Практические занятия проводятся в 6 семестре и позволяют рассмотреть случаи проектирования баз данных на практике и разобрать построение сложных запросов.

Максимальный балл за работу а занятии 5 баллов.

Планируемые результаты при освоении обучающимся 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.

#### **11.4. 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.

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

#### **11.5. 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 данных 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.

#### **11.6. 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 (6 семестр) – 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».

Вопросы и задачи для проведения экзамена представлены в таблице 15.

– зачет (7 семестр) – 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».

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