

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 2022 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 – 2022

Лист согласования рабочей программы дисциплины

Программу составил (а)

старший преподаватель
(должность, уч. степень, звание)



14.06.2022

(подпись, дата)

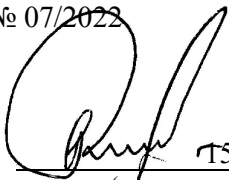
Н.В. Путилова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«15» июня 2022 г, протокол № 07/2022

Заведующий кафедрой № 43

д.т.н., проф.
(уч. степень, звание)



15.06.2022

(подпись, дата)

М.Ю. Охтилев
(инициалы, фамилия)

Ответственный за ОП ВО 09.03.04(02)

Старший преподаватель
(должность, уч. степень, звание)



15.06.2022

(подпись, дата)

А.А. Фоменкова
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

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



15.06.2022

(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. Первый семестр курса посвящен работе с реляционными базами данных, а второй работе с нереляционными базами с упором на их распределенную природу, отличие от распределенных реляционных баз данных. Также дисциплина формирует знания и умения использования баз данных для хранения больших данных.

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 7 зачетных единиц, 252 часа.

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 проектирования баз данных и организации хранения информации в них, работу 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»,
- «Алгоритмы и структуры данных»,
- «Основы программирования».

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

- «Программирование мобильных устройств»
- «Системы искусственного интеллекта»

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 |
| Общая трудоемкость дисциплины, ЗЕ/ (час) | 7/ 252 | 4/ 144 | 3/ 108 |
| Из них часов практической подготовки | | | |
| Аудиторные занятия, всего час. | 40 | 24 | 16 |
| в том числе: | | | |
| лекции (Л), (час) | 16 | 10 | 6 |
| практические/семинарские занятия (ПЗ), (час) | | | |
| лабораторные работы (ЛР), (час) | 24 | 14 | 10 |
| курсовой проект (работа) (КП, КР), (час) | | | |
| экзамен, (час) | 9 | 9 | |
| Самостоятельная работа, всего (час) | 203 | 111 | 92 |
| Вид промежуточной аттестации: 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) |
|---|--------------|---------------|----------|----------|-----------|
| Семестр 6 | | | | | |
| Раздел 1. Моделирование предметной области Тема 1.1 Концептуальное моделирование предметной области Тема 1.2. Логическое и физическое моделирование предметной области Тема 1.3 Нормализация и денормализация баз данных | 3 | | 2 | | 28 |
| Раздел 2. Введение в проектирование баз данных Тема 2.1. История развития и причины появления СУБД. Классификация СУБД. Тема 2.2. Модели данных. Тема 2.3. Место баз данных в программных системах | 1 | | | | 25 |
| Раздел 3. Реляционные базы данных Тема 3.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными Тема 3.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных, манипулирования данными Тема 3.3. Реляционная алгебра | 3 | | 4 | | 25 |

| | | | | | |
|---|----|---|----|---|-----|
| Раздел 4. Введение в язык SQL Тема 4.1. Оператор выборки в языке SQL, агрегатные функции Тема 4.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов Тема 4.3. Использование представлений, управляющих конструкций в языке SQL Тема 4.4. Хранимые процедуры. Назначение хранимых процедур. | 3 | | 7 | | 30 |
| Раздел 5. Администрирование баз данных Тема 5.1. Управление доступом к базе данных. | | | 1 | | 3 |
| Итого в семестре: | 10 | | 14 | | 111 |
| Семестр 7 | | | | | |
| Раздел 4. Введение в язык SQL Тема 4.5. Триггеры. Назначение триггеров: обеспечение активной ссылочной целостности, управление основными данными | 1 | | | | 18 |
| Раздел 5. Администрирование баз данных Тема 5.2 Индексация и оптимизация запросов. Тема 5.3 Целостность баз данных и параллельный доступ Тема 5.4. Выбор типа СУБД применительно к особенностям предметной области | 1 | | | | 14 |
| Раздел 6. Объектные и объектно-реляционные базы данных Тема 6.1. Объектно-реляционные базы данных Тема 6.2. Объектные базы данных | 2 | | 5 | | 30 |
| Раздел 7. NoSQL базы данных Тема 7.1. Базы данных «Ключ-значение» Тема 7.2. Документные базы данных Тема 7.3. Графовые базы данных Тема 7.4. Базы данных «Семейство столбцов» | 2 | | 5 | | 30 |
| Итого в семестре: | 6 | | 10 | | 92 |
| Итого | 16 | 0 | 24 | 0 | 203 |
| | | | | | |

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

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

| Номер раздела | Название и содержание разделов и тем лекционных занятий |
|---------------|---|
| 1 | Раздел 1. Моделирование предметной области Тема 1.1 Концептуальное моделирование предметной области Тема 1.2. Логическое и физическое моделирование предметной области |

| | |
|---|--|
| | Тема 1. 3 Нормализация и денормализация баз данных |
| 2 | Раздел 2. Введение в проектирование баз данных Тема 2.1. История развития и причины появления СУБД. Классификация СУБД. Тема 2.2. Модели данных. Тема 2.3. Место баз данных в программных системах. Базы данных в клиент-серверных приложениях и веб-приложениях. Встроенные базы данных для роботов и интернета вещей и мобильных приложений. Резидентные (in-memory) базы данных для обработки данных в режиме реального времени |
| 3 | Раздел 3. Реляционные базы данных Тема 3.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными Тема 3.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных, манипулирования данными Тема 3.3. Реляционная алгебра |
| 4 | Раздел 4. Введение в язык SQL Тема 4.1. Оператор выборки в языке SQL, агрегатные функции. Применение поиска подстроки и регулярных выражений для обработки текстов как подготовка данных к машинному обучению. Трехзначная логика и обработка NULL-значений в языке SQL Тема 4.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов Тема 4.3. Использование представлений, управляющих конструкций в языке SQL Тема 4.4. Хранимые процедуры, триггеры. Назначение хранимых процедур. Тема 4.5. Триггеры. Назначение триггеров: обеспечение активной ссылочной целостности, управление основными данными |
| 5 | Раздел 5. Администрирование баз данных Тема 5.1 Управление доступом к базе данных. Управление правами доступа к базам данных. Тема 5.2 Индексация и оптимизация запросов. Тема 5.3 Целостность баз данных и параллельный доступ Тема 5.4. Выбор типа СУБД применительно к особенностям предметной области |
| 6 | Раздел 6. Объектные и объектно-реляционные базы данных Тема 6.1. Объектно-реляционные базы данных Тема 6.2. Объектные базы данных |
| 7 | Раздел 7. NoSQL базы данных Тема 7.1. Базы данных «Ключ-значение» Тема 7.2. Документные базы данных Тема 7.3. Графовые базы данных Тема 7.4. Базы данных «Семейство столбцов» |

Все лекции сопровождаются демонстрацией слайдов.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

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

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

| № п/п | Наименование лабораторных работ | Трудоемкость, (час) | Из них практической подготовки, (час) | № раздела дисциплины |
|-----------|---|---------------------|---------------------------------------|----------------------|
| Семестр 6 | | | | |
| 1 | Разработка физической модели базы данных с учетом декларативной ссылочной целостности | 2 | | 1 |
| 2 | Создание и модификация базы данных и таблиц базы данных | 2 | | 3 |
| 3 | Заполнение таблиц и модификация данных | 2 | | 3 |
| 4 | Разработка SQL запросов: виды соединений и шаблоны | 2 | | 4 |
| 5 | Разработка SQL запросов: запросы с подзапросами | 3 | | 4 |
| 6 | Хранимые процедуры. Управление доступом | 3 | | 4,5 |
| Семестр 7 | | | | |
| 7 | Объектно-реляционные базы данных. Проектирование и создание | 3 | | 6 |
| 8 | Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы | 2 | | 6 |
| 9 | Разработка документной базы данных | 3 | | 7 |
| 10 | Манипулирование данными в документной базе данных | 2 | | 7 |
| Всего | | 24 | | |

4.5. Курсовое проектирование/ выполнение курсовой работы
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

| Вид самостоятельной работы | Всего, час | Семестр 6, час | Семестр 7, час |
|---|------------|----------------|----------------|
| 1 | 2 | 3 | 4 |
| Изучение теоретического материала дисциплины (ТО) | 134 | 60 | 74 |
| Контрольные работы заочников (КРЗ) | 16 | 8 | 8 |
| Подготовка к промежуточной аттестации (ПА) | 53 | 43 | 10 |
| Всего: | 203 | 111 | 92 |

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий
Перечень печатных и электронных учебных изданий приведен в таблице 8.
Таблица 8– Перечень печатных и электронных учебных изданий

| Шифр/ URL адрес | Библиографическая ссылка | Количество экземпляров в библиотеке (кроме электронных экземпляров) |
|--|--|---|
| https://e.lanbook.com/book/193373 — Режим доступа: для авториз. пользователей | Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 244 с. — ISBN 978-5-8114-9368-5. — Текст : электронный | |
| https://e.lanbook.com/book/176572 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. | |
| https://znanium.com/catalog/product/1227737 – 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. | |
| https://znanium.com/catalog/product/1834412 Режим доступа: по подписке. | Дадян, Э. Г. Методы, модели, средства хранения и обработки данных : учебник / Э.Г. Дадян, Ю.А. Зеленков. — Москва : Вузовский учебник : ИНФРА-М, 2022. — 168 с. - ISBN 978-5-9558-0490-3. - Текст : электронный. - | |
| https://znanium.com/catalog/product/1214862 Режим доступа: по | Мартишин, С. А. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с | |

| | | |
|--|---|--|
| подписке. | использованием Node.js, Express.js, Apache Spark и Scala : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ИНФРА-М, 2021. — 235 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cc063e18baca3.52928692. - ISBN 978-5-16-015133-5. - Текст : электронный. | |
| https://znanium.com/catalog/product/1215513 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. | |
| https://znanium.com/catalog/product/1073525 Режим доступа: по подписке | Григорьев, А. А. Передача, хранение и обработка больших объемов научных данных : учебное пособие / А.А. Григорьев, Е.А. Исаев, П.А. Тарасов. — Москва : ИНФРА-М, 2021. — 207 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1073525. - ISBN 978-5-16-015985-0. - Текст : электронный. | |
| https://znanium.com/catalog/product/1308393 Режим доступа: по подписке. | Игнатъев, В. В. Методы управления техническими объектами с помощью интеллектуальных регуляторов на основе самоорганизации баз знаний : монография / В. В. Игнатъев ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2020. - 142 с. - ISBN 978-5-9275-3562-0. - 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 |
|---|--|
| http://citforum.ru/database/ | Базы данных |
| https://dev.mysql.com/doc/ | Электронная документация по MySQL |
| http://www.mysql.ru/docs/man/Reference.html | Электронная документация по MySQL на русском языке |

| | |
|---|---|
| https://docs.mongodb.com/ | Электронная документация по MongoDB |
| https://postgrespro.ru/docs/postgresql | Электронная документация по PostgreSQL на русском языке |
| https://hbase.apache.org/ | Электронная документация по HBASE |
| https://www.ibm.com/docs/en/db2/11.5?topic=learning-machine-stored-procedures | Перечень хранимых процедур СУБД DB2 для машинного обучения |
| https://habr.com/ru/ | Хабр — русскоязычный веб-сайт тематических коллективных для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом. |
| https://habr.com/ru/hub/db_admins/ | Канал Хабр о базах данных |
| https://habr.com/ru/hub/bigdata/ | Канал Хабр о больших данных |

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 | Draw.io |
| 2 | MySQL Community Server |
| 3 | Microsoft Word или OpenOffice Writer/ LibreOffice Writer |
| 5 | MongoDB Community Server |
| 6 | PostgreSQL |
| 7 | MySQL Workbench |

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

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

| № п/п | Наименование |
|-------|------------------|
| | Не предусмотрено |

9. Материально-техническая база

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

Таблица 12 – Состав материально-технической базы

| № п/п | Наименование составной части материально-технической базы | Номер аудитории (при необходимости) |
|-------|---|-------------------------------------|
| 1 | Мультимедийная лекционная аудитория | — |
| 2 | 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

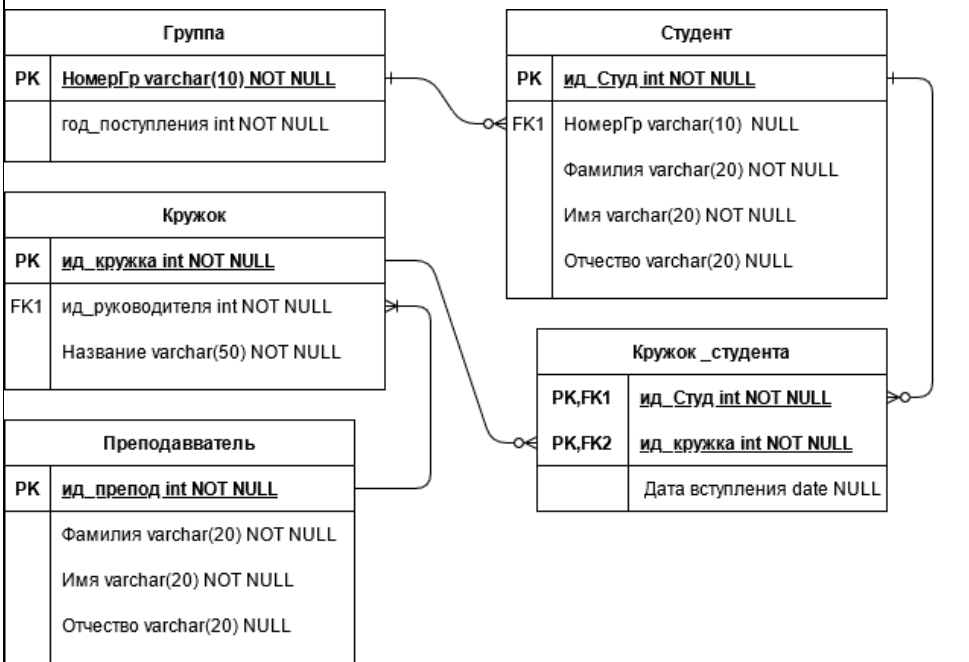
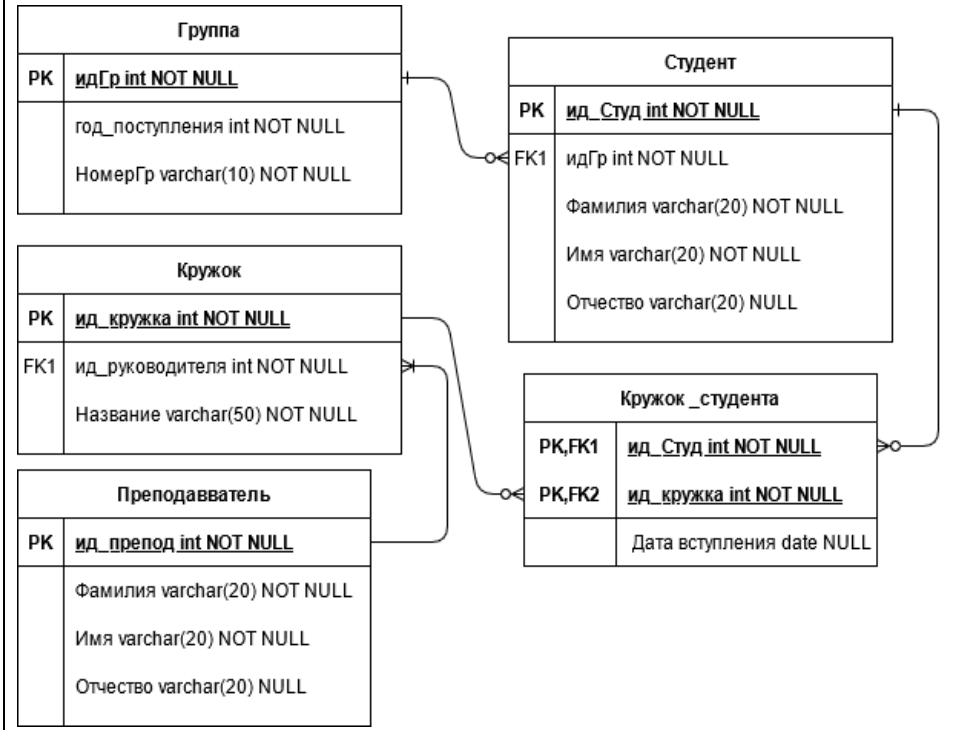
| Оценка компетенции | Характеристика сформированных компетенций |
|---------------------------------------|---|
| 5-балльная шкала | |
| «отлично» «зачтено» | <ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно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

| | |
|--------------------|--|
| Оценка компетенции | Характеристика сформированных компетенций |
| 5-балльная шкала | |
| | – не может аргументировать научные положения; – не формулирует выводов и обобщений. |

10.3. Типовые контрольные задания или иные материалы.
Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

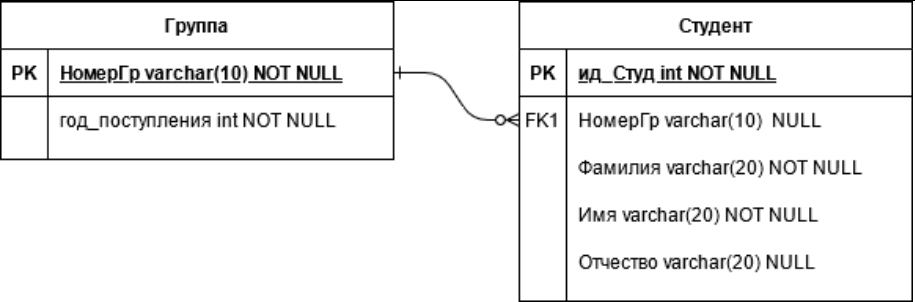
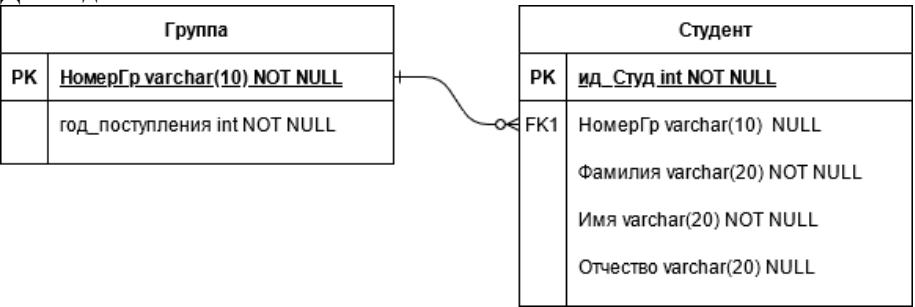
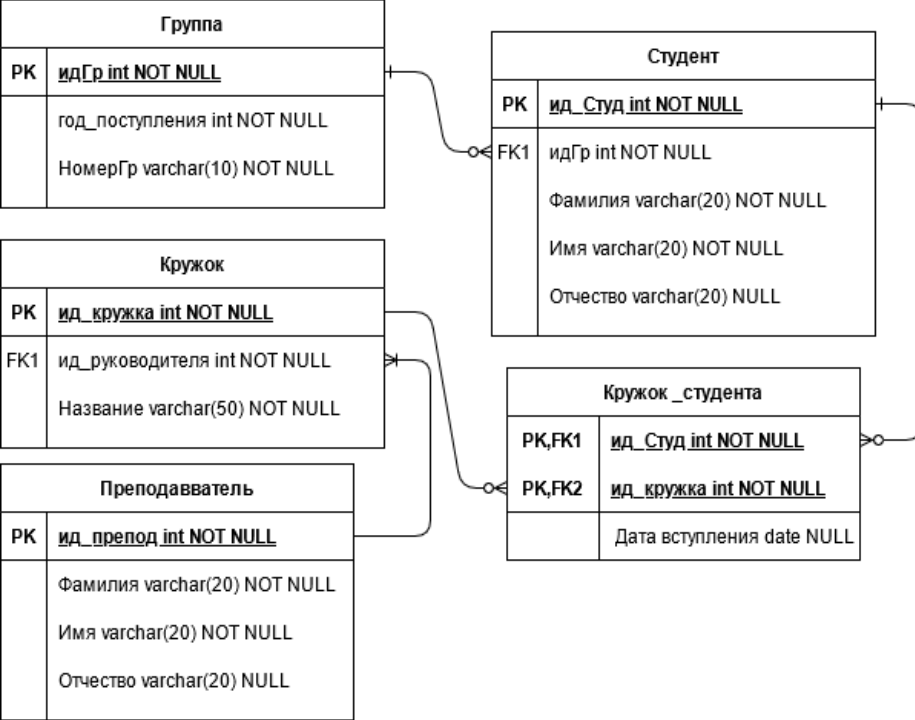
| № п/п | Перечень вопросов (задач) для экзамена | Код индикатора |
|-------|--|----------------|
| № п/п | Перечень вопросов (задач) для экзамена | Код индикатора |
| 1 | Концептуальное проектирование баз данных | ОПК-6.3.1 |
| 2 | Диаграмма «сущность-связь» | ОПК-6.3.1 |
| 3 | Логическое и физическое проектирование баз данных . | ОПК-6.3.1 |
| 4 | История причины появления СУБД. Классификация СУБД. | ОПК-6.3.1 |
| 5 | История развития СУБД и модели данных. | ОПК-6.3.1 |
| 6 | Объектно-ориентированные и NoSQL модели данных. | ОПК-6.3.1 |
| 7 | Нормальные формы 1-3 и денормализация | ОПК-6.3.1 |
| | Нормальные формы Бойса-Кодда и 4-6 | ОПК-6.3.1 |
| 8 | Отношения и их свойства, ключи отношений | ОПК-6.3.1 |
| 9 | Типы данных в языке SQL, оператор языка SQL для создания таблиц | ОПК-6.3.1 |
| 10 | Операторы языка SQL для удаления и модификации таблиц | ОПК-6.3.1 |
| 11 | Абстрактные операции манипулирования данными, их реализация в языке SQL | ОПК-6.3.1 |
| 12 | Булевы операции над отношениями | ОПК-6.3.1 |
| 13 | Операция выборки в реляционной алгебре и ее свойства | ОПК-6.3.1 |
| 14 | Операция проекции в реляционной алгебре и, ее свойства | ОПК-6.3.1 |
| 15 | Операции соединения (естественного, эквисоединения и θ -соединения) в реляционной алгебре и ее свойства | ОПК-6.3.1 |
| 16 | Операция деления в реляционной алгебре | ОПК-6.3.1 |
| 17 | Операция переименования атрибутов в реляционной алгебре | ОПК-6.3.1 |
| 18 | Оператор выборки в языке SQL | ОПК-6.3.1 |
| 19 | Агрегатные функции в операторе выборки языка SQL | ОПК-6.3.1 |
| 20 | Объединение, пересечение, разность запросов в языке SQL | ОПК-6.3.1 |
| 21 | Запросы с подзапросами в языке SQL | ОПК-6.3.1 |
| 22 | Экзистенциальные запросы в языке SQL | ОПК-6.3.1 |
| 23 | Виды соединений в языке SQL | ОПК-6.3.1 |
| 24 | Трехзначная логика и обработка NULL-значений в языке SQL | ОПК-6.3.1 |
| 25 | Представления в языке SQL | ОПК-6.3.1 |
| 26 | Управляющие конструкции в языке SQL | ОПК-6.3.1 |
| 27 | Хранимые процедуры в языке SQL | ОПК-6.3.1 |
| 28 | Управление доступом в базах данных | ОПК-6.3.1 |
| 29 | Место баз данных в программных системах | ОПК-6.3.1 |
| | Задачи | |
| 1. | Написать скрипт SQL, создающий таблицу из 2 столбцов, один из которых является строкой фиксированной длины и первичным ключом, | ОПК-6.У.1 |

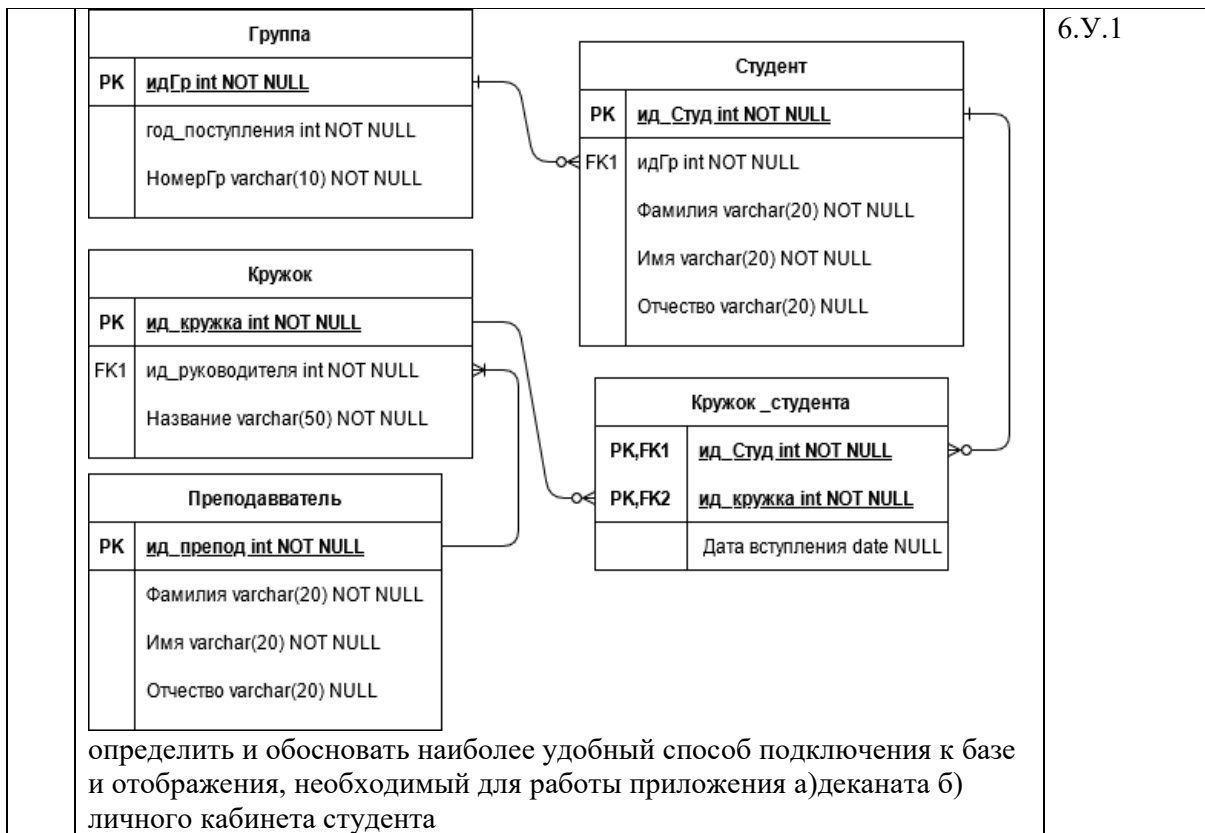
| | | |
|----|---|-----------|
| 2. | <p>а второй строкой ограниченной длины и внешним ключом</p> <p>Написать скрипт SQL, создающий базу данных по заданной схеме:</p>  <pre> graph TD subgraph "Группа" GPK[PK НомерГр varchar(10) NOT NULL] G1[год_поступления int NOT NULL] end subgraph "Студент" SPK[PK ид_Студ int NOT NULL] S1[НомерГр varchar(10) NULL] S2[Фамилия varchar(20) NOT NULL] S3[Имя varchar(20) NOT NULL] S4[Отчество varchar(20) NULL] end subgraph "Кружок" CKPK[PK ид_кружка int NOT NULL] CKFK1[FK1 ид_руководителя int NOT NULL] CK1[Название varchar(50) NOT NULL] end subgraph "Преподаватель" PK1[PK ид_препод int NOT NULL] P1[Фамилия varchar(20) NOT NULL] P2[Имя varchar(20) NOT NULL] P3[Отчество varchar(20) NULL] end subgraph "Кружок_студента" CSFK1[PK,FK1 ид_Студ int NOT NULL] CSFK2[PK,FK2 ид_кружка int NOT NULL] CS1[Дата вступления date NULL] end GPK --> S1 CKPK --> S1 CKFK1 --> P1 CKFK1 --> P2 CKFK1 --> P3 CSFK1 --> SPK CSFK2 --> CKPK </pre> | ОПК-6.У.1 |
| 3. | <p>Написать скрипт SQL, вставляющий данные в базу данных по заданной схеме:</p>  <pre> graph TD subgraph "Группа" GPK[PK идГр int NOT NULL] G1[год_поступления int NOT NULL] G2[НомерГр varchar(10) NOT NULL] end subgraph "Студент" SPK[PK ид_Студ int NOT NULL] S1[идГр int NOT NULL] S2[Фамилия varchar(20) NOT NULL] S3[Имя varchar(20) NOT NULL] S4[Отчество varchar(20) NULL] end subgraph "Кружок" CKPK[PK ид_кружка int NOT NULL] CKFK1[FK1 ид_руководителя int NOT NULL] CK1[Название varchar(50) NOT NULL] end subgraph "Преподаватель" PK1[PK ид_препод int NOT NULL] P1[Фамилия varchar(20) NOT NULL] P2[Имя varchar(20) NOT NULL] P3[Отчество varchar(20) NULL] end subgraph "Кружок_студента" CSFK1[PK,FK1 ид_Студ int NOT NULL] CSFK2[PK,FK2 ид_кружка int NOT NULL] CS1[Дата вступления date NULL] end GPK --> S1 CKPK --> S1 CKFK1 --> P1 CKFK1 --> P2 CKFK1 --> P3 CSFK1 --> SPK CSFK2 --> CKPK </pre> | ОПК-6.У.1 |
| 4. | <p>Написать скрипт SQL, удаляющий кружки, в которых нет студентов данные из базы данных по заданной схеме:</p> | ОПК-6.У.1 |

| | | |
|----|--|-----------|
| | <p>The ER diagram consists of five tables:</p> <ul style="list-style-type: none"> Группа: Primary key <u>идГр int NOT NULL</u>. Attributes: год_поступления int NOT NULL, НомерГр varchar(10) NOT NULL. Студент: Primary key <u>ид_Студ int NOT NULL</u>. Attributes: идГр int NOT NULL (FK1), Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL. Кружок: Primary key <u>ид_кружка int NOT NULL</u>. Attributes: ид_руководителя int NOT NULL (FK1), Название varchar(50) NOT NULL. Преподаватель: Primary key <u>ид_препод int NOT NULL</u>. Attributes: Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL. Кружок_студента: Composite primary key <u>ид_Студ int NOT NULL</u> (PK,FK1) and <u>ид_кружка int NOT NULL</u> (PK,FK2). Attribute: Дата вступления date NULL. <p>Relationships:</p> <ul style="list-style-type: none"> Группа (1) to Студент (N): 1:M relationship. Кружок (1) to Преподаватель (N): 1:M relationship. Кружок (1) to Студент (N): 1:M relationship. Кружок (1) to Кружок_студента (N): 1:M relationship. Студент (N) to Кружок_студента (N): 1:M relationship. | |
| 5. | <p>Написать запрос SQL, выбирающий кружок с количеством студентов больше среднего по заданной схеме:</p> | ОПК-6.У.1 |
| 6. | <p>Написать запрос SQL, выбирающий кружок , в который ходят все студенты группы 4136 по заданной схеме:</p> | ОПК-6.У.1 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------|--|-----------|--------------------------|--|-------------------------------|--|------------------------------|----|-------------------------------|-----|------------------------------|--|-------------------------------|----|-------------------------------|--|------------------------------|--|--------------------------|--|---------------------------|----|-----------------------------|-----|-------------------|--|------------------------------|--|--------------------------|--|---------------------------|--------|-----------------------------|--------|-------------------------------|--|---------------------------|--|
| | <p>Группа</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>Кружок</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>Преподаватель</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>Студент</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>Кружок_студента</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_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 | <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,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_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 | <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,FK1 | <u>ид_Студ int NOT NULL</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PK,FK2 | <u>ид_кружка int NOT NULL</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Дата вступления date NULL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8. | <p>Для заданной схемы разработать хранимую процедуру, осуществляющую каскадное удаление группы и всех её студентов:</p> | ОПК-6.У.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | |
|-----|---|-----------|
| | <p>The diagram shows the following tables and relationships:</p> <ul style="list-style-type: none"> Группа: PK <u>идГр int NOT NULL</u>, год_поступления int NOT NULL, НомерГр varchar(10) NOT NULL Студент: PK <u>ид_Студ int NOT NULL</u>, FK1 идГр int NOT NULL, Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL Кружок: PK <u>ид_кружка int NOT NULL</u>, FK1 ид_руководителя int NOT NULL, Название varchar(50) NOT NULL Преподаватель: PK <u>ид_препод int NOT NULL</u>, Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL Кружок_студента: PK,FK1 <u>ид_Студ int NOT NULL</u>, PK,FK2 <u>ид_кружка int NOT NULL</u>, Дата вступления date NULL <p>Relationships:</p> <ul style="list-style-type: none"> Группа (1) to Студент (0..1) via FK1 Кружок (1) to Преподаватель (0..1) via FK1 Кружок (1) to Студент (0..1) via FK1 Кружок_студента (0..1) to Студент (1) via PK,FK1 Кружок_студента (0..1) to Кружок (1) via PK,FK2 | |
| 9. | <p>Для заданной схемы разработать хранимую процедуру, осуществляющую логическое удаление группы и всех её студентов:</p> | ОПК-6.У.1 |
| 10. | Для заданной схемы | ОПК-6.У.1 |

| | | |
|-----|--|-----------|
| |  <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> | |
| 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>Добавить пользователя Perfect в базу данных «upi» дать ему право на изменение только таблицы «Группа», а остальных таблиц права на чтение.</p> | ОПК-6.У.1 |
| 13. | <p>Для базы данных по заданной схеме</p>  <p>определить и обосновать уровень изоляции транзакций, необходимый для работы приложения а) деканата б) личного кабинета студента</p> | ОПК-6.У.1 |
| 14. | <p>Для базы данных по заданной схеме</p> | ОПК- |



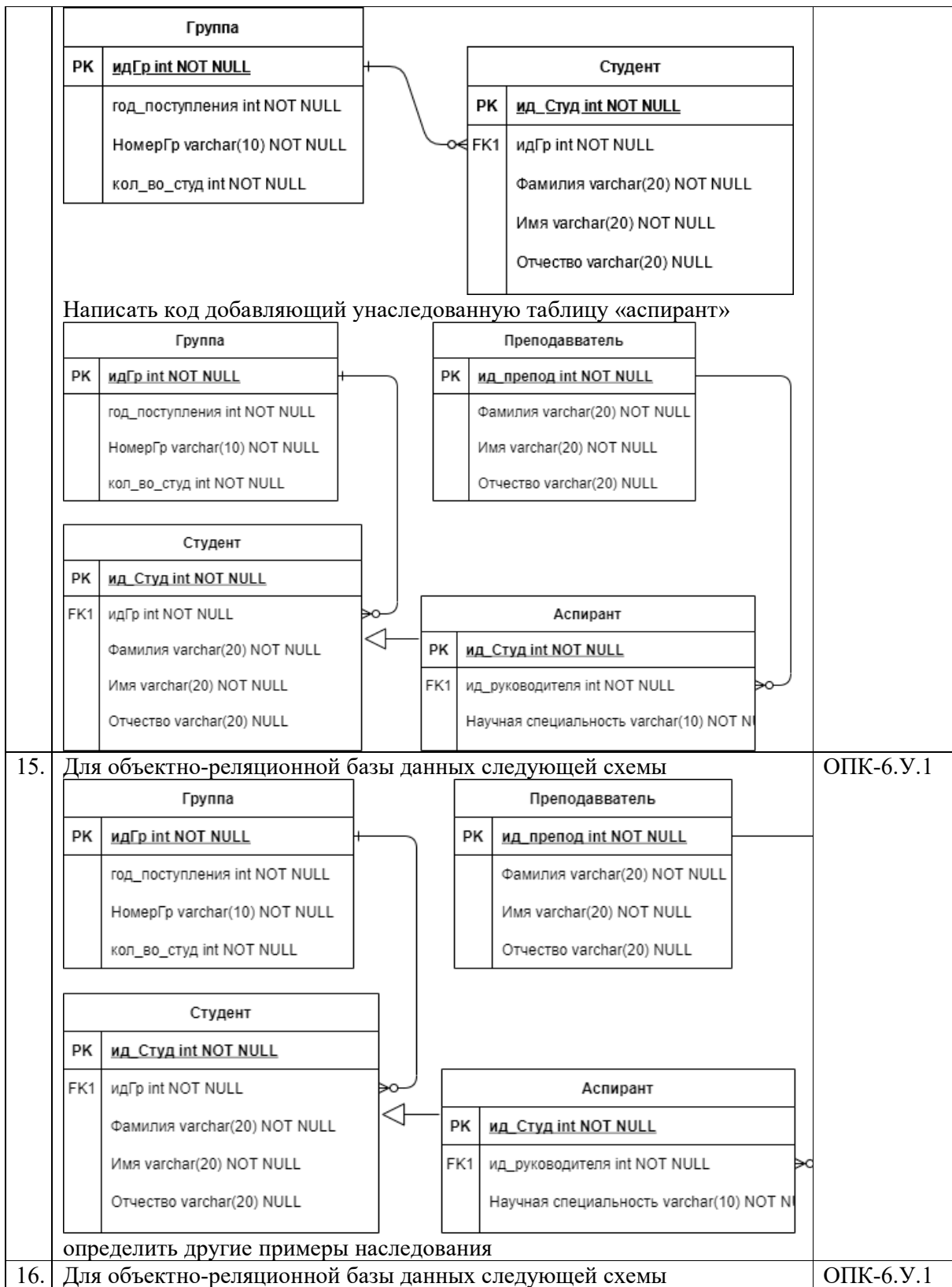
Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.
Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

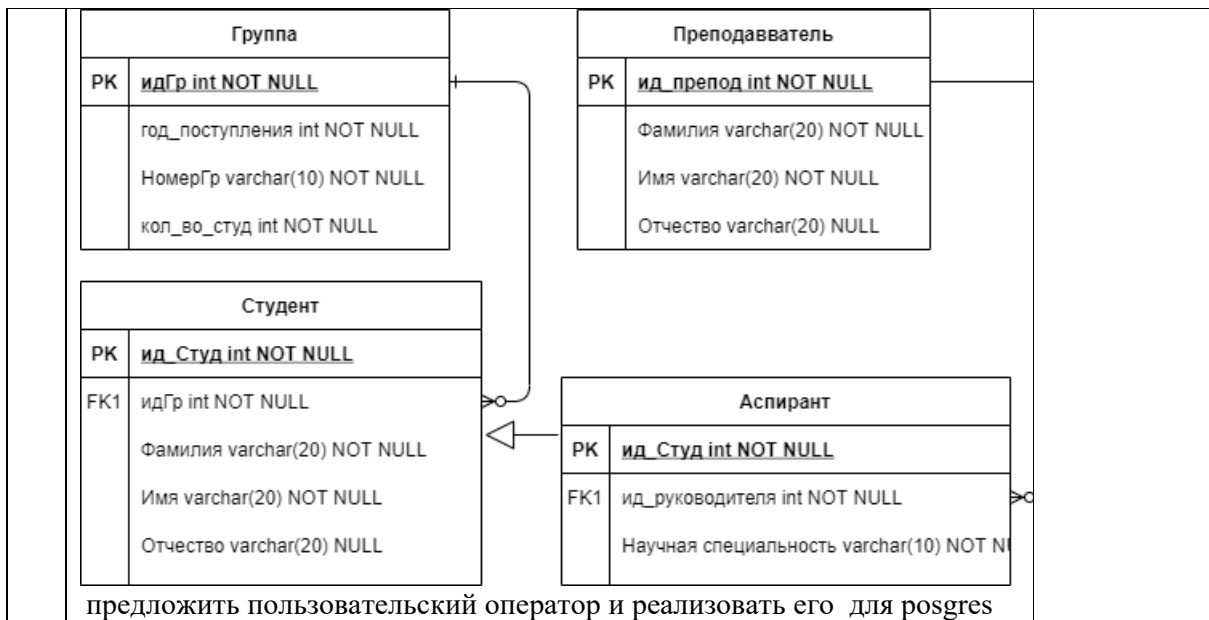
| № п/п | Перечень вопросов (задач) для зачета / дифф. зачета | Код индикатора |
|-------|---|----------------|
| 1 | Триггеры в языке SQL | ОПК-6.3.1 |
| 2 | Индексация данных | ОПК-6.3.1 |
| 3 | Целостность баз данных и параллельный доступ | ОПК-6.3.1 |
| 4 | Выбор типа СУБД применительно к особенностям предметной области | ОПК-6.3.1 |
| 5 | Объектные базы данных. (Объектная модель). Проектирование и создание | ОПК-6.3.1 |
| 6 | Объектные базы данных. Манипуляции данными | ОПК-6.3.1 |
| 7 | Объектно-реляционные базы данных. Проектирование и создание | ОПК-6.3.1 |
| 8 | Объектно-реляционные базы данных. Пользовательские типы и операторы | ОПК-6.3.1 |
| 9 | Базы данных «Ключ-значение» | ОПК-6.3.1 |
| 10 | Структура и создание документных баз данных. | ОПК-6.3.1 |
| 11 | Манипуляции данными и поиск в документных базах данных. | ОПК-6.3.1 |
| 12 | Графовые базы данных. Графовые структуры, назначение графовых бд | ОПК-6.3.1 |
| 13 | Графовые базы данных: графы и операции CRUD | ОПК-6.3.1 |
| 14 | Базы данных типа «Семейство столбцов». Операции CRUD и администрирование таблиц | ОПК-6.3.1 |
| 15 | Базы данных типа «Семейство столбцов», работа с «большими данными» | |
| | Задачи | |
| 1. | Для базы данных по заданной схеме написать триггеры, поддерживающие в актуальном состоянии значение столбца | ОПК-6.У.1 |

| | | |
|----|---|-----------|
| | <p>кол во студ, содержащего количество студентов группы.</p> | |
| 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. | <p>Добавить пользователя Perfect в базу данных «uni» дать ему право на изменение только таблицы «Группа», а остальных таблиц права на чтение.</p> | ОПК-6.У.1 |
| 4. | <p>Для базы данных по заданной схеме</p> <p>определить и обосновать уровень изоляции транзакций, необходимый для работы приложения а) деканата б) личного кабинета студента</p> | ОПК-6.У.1 |
| 5. | <p>Для предметной области «учебный план» (дисциплины, кафедры,</p> | ОПК-6.У.1 |

| | | |
|----|--|-----------|
| | знания и умения полученные в результате освоения дисциплин) предложить и обосновать выбор модели данных СУБД. Предложить примерную схему базы данных. | |
| 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. | <p>Для реляционной базы со схемой</p> <p>The diagram shows the following tables and relationships:</p> <ul style="list-style-type: none"> Группа (Group): PK <u>идГр int NOT NULL</u>, год_поступления int NOT NULL, НомерГр varchar(10) NOT NULL. Студент (Student): PK <u>ид_Студ int NOT NULL</u>, FK1 <u>идГр int NOT NULL</u>, Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL. Кружок (Club): PK <u>ид_кружка int NOT NULL</u>, FK1 <u>ид_руководителя int NOT NULL</u>, Название varchar(50) NOT NULL. Преподаватель (Teacher): PK <u>ид_препод int NOT NULL</u>, Фамилия varchar(20) NOT NULL, Имя varchar(20) NOT NULL, Отчество varchar(20) NULL. Кружок_студента (Student-Club): PK,FK1 <u>ид_Студ int NOT NULL</u>, PK,FK2 <u>ид_кружка int NOT NULL</u>, Дата вступления date NULL. <p>Relationships:</p> <ul style="list-style-type: none"> Группа (1) to Студент (N): 1:M relationship. Кружок (1) to Студент (N): 1:M relationship. Преподаватель (1) to Кружок (N): 1:M relationship. Кружок_студента (1) to Студент (N): 1:M relationship. Кружок_студента (1) to Кружок (N): 1:M relationship. <p>создать соответствующую ей (по данным) документную базу данных</p> | ОПК-6.У.1 |
| 9. | Для реляционной базы со схемой | ОПК-6.У.1 |

| | | |
|-----|--|-----------|
| | <p>создать соответствующую ей (по данным) базу данных «ключ-значение»</p> | |
| 10. | 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. | ОПК-6.У.1 |
| 11. | Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт создания /добавления данных для MongoDB | ОПК-6.У.1 |
| 12. | Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт выбирающий студентов, сдавших наибольшее количество работ по дисциплине «Проектирование баз данных» | ОПК-6.У.1 |
| 13. | Для документной базы данных следующей схемы (коллекция студентов, для каждого из которых заданы фамилия, имя, отчество (не обязательно), список его работ, для каждой работы задано название, содержание(текст), дата прикрепления, дисциплина, статус работы) написать скрипт удаляющий все работы, прикрепленные в позапрошлом году | ОПК-6.У.1 |
| 14. | Для объектно-реляционной базы данных следующей схемы | 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. | 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. Графовая | ОПК-6.3.1 |
| 2. | Назовите описываемую модель данных одним словом. Модель данных, реализующая структуру данных «дерево» с узлом, содержащим изначально жестко заданную табличную структуру. Иерархическая | ОПК-6.3.1 |
| 3. | 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. Сетевая | ОПК-6.3.1 |

| | | |
|-----|--|-----------|
| | | |
| 4. | <p>Назовите описываемую модель данных одним словом.</p> <p>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>Документная</p> | ОПК-6.3.1 |
| 5. | <p>Продолжите фразу:</p> <p>Домен атрибута — это</p> <p>а. диапазон допустимых значений атрибута</p> <p>б. -тип данных атрибута</p> <p>в. множество значений атрибута, находящихся в данный момент в теле отношения</p> <p>г. множество атрибутов, объединенных одним типом данных</p> | ОПК-6.3.1 |
| 6. | <p>Продолжите фразу:</p> <p>В отношении одинаковые строки</p> <p>а. могут существовать</p> <p>б. не могут существовать</p> <p>в. могут возникнуть как результаты операций реляционной алгебры</p> <p>г. могут существовать в ненормализованной базе данных</p> | ОПК-6.3.1 |
| 7. | <p>Какой ключ описывается такими словами (ответьте одним словом)?</p> <p>Потенциальный ключ, выбранный для уникальной идентификации всех остальных значения атрибутов в любой строке.</p> <p>Первичный</p> | ОПК-6.3.1 |
| 8. | <p>«Отношение без зависимостей соединения.»</p> <p>Это требование какой нормальной формы?</p> <p>Пятая нормальная форма</p> | ОПК-6.3.1 |
| 9. | <p>«Каждый детерминант отношения является потенциальным ключом.»</p> <p>Это требование какой нормальной формы?</p> <p>Нормальная форма Бойса –Кодда</p> | ОПК-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</p> <p>б. Клиентские</p> <p>в. Встроенные</p> <p>г. Резидентные</p> | ОПК-6.3.1 |
| 11. | <p>Дана предметная область</p> | ОПК-6.3.1 |

| | <p>R1</p> <table border="1" data-bbox="342 184 987 296"> <tr> <th>Студент</th> <th>дисциплина</th> <th>тема курсового</th> <th>оценка</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p>R2</p> <table border="1" data-bbox="342 365 950 476"> <tr> <th>группа</th> <th>дисциплина</th> <th>вид занятия</th> <th>семестр</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p>R3</p> <table border="1" data-bbox="342 575 621 653"> <tr> <th>Студент</th> <th>группа</th> </tr> <tr> <td></td> <td></td> </tr> </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> <ol style="list-style-type: none"> Студенты, у которых есть лабораторные занятия во всех семестрах Семестры, в которых у всех студентов есть лабораторные занятия Все студенты, у которых есть лабораторные занятия в семестре Все семестры, в которых у студентов есть лабораторные занятия | Студент | дисциплина | тема курсового | оценка | | | | | группа | дисциплина | вид занятия | семестр | | | | | Студент | группа | | | |
|---------|---|-------------------|----------------|-------------------|----------------|------------------|---|--------|------|--------|------------|----------------|---------|------|------|-------|---|---------|--------|-------|------|-----------|
| 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?</p> <p>Коррелированный</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> <table border="1" data-bbox="354 1371 1243 1528"> <thead> <tr> <th>id_s</th> <th>surname</th> <th>name</th> <th>is_red_diploma</th> <th>is_special_grant</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Иванов</td> <td>Иван</td> <td>true</td> <td>null</td> </tr> <tr> <td>2</td> <td>Петров</td> <td>Петр</td> <td>null</td> <td>false</td> </tr> <tr> <td>3</td> <td>Сидоров</td> <td>Сидор</td> <td>false</td> <td>true</td> </tr> </tbody> </table> <p>Какой результат выдаст запрос ниже?</p> <pre>SELECT is_red_diploma AND is_special_grant FROM student WHERE id_s=1</pre> <p>Null</p> | 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 | 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 | | | | | | | | | | | | | | | | | | |
| 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>Замена представления</p> | ОПК-6.3.1 | | | | | | | | | | | | | | | | | | | | |
| 15. | <p>К таким переменным можно получить доступ без объявления, им</p> | ОПК-6.3.1 | | | | | | | | | | | | | | | | | | | | |

| | | |
|-----|--|-----------|
| | <p>может быть присвоено значение из ограниченного набора типов данных они зависят от сеанса, имея при этом сквозную видимость.</p> <ul style="list-style-type: none"> а. Пользовательские переменные б. Сеансовые переменные в. Локальные переменные г. Системные переменные | |
| 16. | <p>Какой тип данных является числовым типом с фиксированной запятой?</p> <p>Numeric Decimal</p> | ОПК-6.3.1 |
| 17. | <p>Какие уровни изоляции транзакций защищают от фантомных (строк)?</p> <ul style="list-style-type: none"> а. SERIALIZABLE б. READ UNCOMMITTED в. READ COMMITTED г. REPEATABLE READ | ОПК-6.3.1 |
| 18. | <p>Как расшифровывается термин B*-дерево, являющееся одним из типов индексов по организации хранения ?</p> <ul style="list-style-type: none"> а. Сбалансированное дерево б. Бинарное дерево в. Двухуровневое дерево г. Сильноветвящееся дерево | 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> <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> <ul style="list-style-type: none"> а. 3 б. 1 в. Неизвестно, так как неизвестна корректность вставляемых данных | ОПК-6.3.1 |
| 20. | <p>Выберите все корректные варианты из предложенных</p> <p>В операторе создания таблиц можно</p> <ul style="list-style-type: none"> а. указать способы поддержания ссылочной целостности б. указать внешний ключ таблицы в. указать значение по умолчанию для столбца | ОПК-6.У.1 |

| | <p>г. указать вторичный ключ в таблицы</p> <p>д. создать домен столбца</p> <p>е. назначить псевдоним по умолчанию для таблицы</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|---------|----------|--|--|--------------|-----|------|----------|----------|-------------|------|----------|---------|-------------|--|------|----------|-------------|--|----------|----------|-------------|--|------|-------------|--|--|--|-------------|-----|------|----------|---------|-------------|--|----------|------|-------------|--|------|----------|-------------|--|------|--------|-------------|--|------|-----------|
| 21. | <p>Какой или какие из операторов корректно добавляют столбец к заполненной таблице</p> <table border="1"> <thead> <tr> <th colspan="4">student</th> </tr> </thead> <tbody> <tr> <td><u>id_st</u></td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>id_group</td> <td>int</td> <td><fk></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;</p> <p>б. ALTER TABLE student ADD COLUMN email varchar(15) null;</p> <p>в. ALTER TABLE student ADD email varchar(15) not null;</p> <p>г. ADD TO student email varchar(15) null;</p> | student | | | | <u>id_st</u> | 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>id_st</u> | 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"> <thead> <tr> <th colspan="4">student</th> </tr> </thead> <tbody> <tr> <td><u>id_st</u></td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>id_group</td> <td>int</td> <td><fk></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');</p> <p>б. INSERT INTO student VALUES (1, 1, 'Иванов', 'Иван', 'Иванович');</p> <p>в. INSERT INTO student (id_st, surname) VALUES (1, 'Иванов');</p> <p>г. INSERT INTO student (1, 1, 'Иванов', 'Иван', 'Иванович');</p> | student | | | | <u>id_st</u> | 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>id_st</u> | 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> <table border="1"> <thead> <tr> <th colspan="4">Teacher</th> </tr> </thead> <tbody> <tr> <td><u>id_T</u></td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>surname</td> <td>varchar(25)</td> <td></td> <td>not null</td> </tr> <tr> <td>name</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> <tr> <td>patronym</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> <tr> <td>degree</td> <td>varchar(30)</td> <td></td> <td>null</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">new_Teacher</th> </tr> </thead> <tbody> <tr> <td><u>id_T</u></td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>surname</td> <td>varchar(25)</td> <td></td> <td>not null</td> </tr> <tr> <td>name</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> <tr> <td>patronym</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> <tr> <td>degree</td> <td>varchar(30)</td> <td></td> <td>null</td> </tr> </tbody> </table> <p>Какой или какие из операторов полностью корректно приводит данные таблицы Teacher в соответствии с данными таблицы new_Teacher?</p> <p>а.</p> <p>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=</p> | Teacher | | | | <u>id_T</u> | int | <pk> | not null | surname | varchar(25) | | not null | name | varchar(25) | | null | patronym | varchar(25) | | null | degree | varchar(30) | | null | new_Teacher | | | | <u>id_T</u> | int | <pk> | not null | surname | varchar(25) | | not null | name | varchar(25) | | null | patronym | varchar(25) | | null | degree | varchar(30) | | null | ОПК-6.У.1 |
| Teacher | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>id_T</u> | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| surname | varchar(25) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| name | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| patronym | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| degree | varchar(30) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| new_Teacher | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>id_T</u> | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| surname | varchar(25) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| name | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| patronym | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| degree | varchar(30) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```

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
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)

```

24.

Какой или какие из операторов полностью корректно удалят из изображенной ниже таблицы студентов с идентификатором большим тысячи?

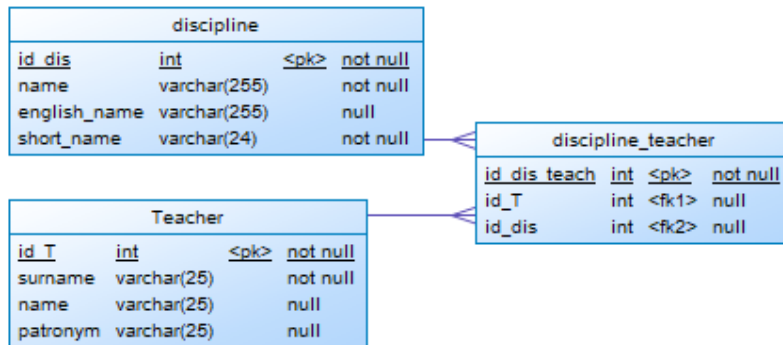
| student | | | |
|----------|-------------|------|----------|
| id_st | int | <pk> | not null |
| id_group | int | <fk> | null |
| surname | varchar(20) | | null |
| name | varchar(20) | | not null |
| patronym | varchar(20) | | null |

- a. DELETE FROM student WHERE id_st>1000;
- б. DELETE FROM student WHERE not(id_st<=1000);
- в. DELETE student WHERE id_st>1000;

ОПК-6.У.1

| | <p>г. TRUNCATE TABLE student WHERE id_st>1000;</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|--|------------|----------|--|--|--------|-----|------|----------|--------|--------------|--|----------|------------------|--------------|--|------|------------|-------------|--|----------|---------|--|--|--|------|-----|------|----------|---------|-------------|--|----------|------|-------------|--|------|----------|-------------|--|------|--------------------|--|--|--|--------------|-----|------|----------|------|-----|-------|------|--------|-----|-------|------|------------------|
| <p>25.</p> | <p>Существует таблица групп,</p> <table border="1" data-bbox="347 247 686 344"> <thead> <tr> <th colspan="4">st_group</th> </tr> </thead> <tbody> <tr> <td>id_gr</td> <td>int</td> <td><pk></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 | <p>ОПК-6.У.1</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| st_group | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_gr | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| num_gr | varchar(10) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>26.</p> | <p>Дана предметная область , описывающая ведение дисциплин (discipline) преподавателями (teacher).</p> <table border="1" data-bbox="407 1094 821 1241"> <thead> <tr> <th colspan="4">discipline</th> </tr> </thead> <tbody> <tr> <td>id_dis</td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>name</td> <td>varchar(255)</td> <td></td> <td>not null</td> </tr> <tr> <td>english_name</td> <td>varchar(255)</td> <td></td> <td>null</td> </tr> <tr> <td>short_name</td> <td>varchar(24)</td> <td></td> <td>not null</td> </tr> </tbody> </table> <table border="1" data-bbox="407 1283 764 1430"> <thead> <tr> <th colspan="4">Teacher</th> </tr> </thead> <tbody> <tr> <td>id_T</td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>surname</td> <td>varchar(25)</td> <td></td> <td>not null</td> </tr> <tr> <td>name</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> <tr> <td>patronym</td> <td>varchar(25)</td> <td></td> <td>null</td> </tr> </tbody> </table> <table border="1" data-bbox="873 1209 1187 1329"> <thead> <tr> <th colspan="4">discipline_teacher</th> </tr> </thead> <tbody> <tr> <td>id_dis_teach</td> <td>int</td> <td><pk></td> <td>not null</td> </tr> <tr> <td>id_T</td> <td>int</td> <td><fk1></td> <td>null</td> </tr> <tr> <td>id_dis</td> <td>int</td> <td><fk2></td> <td>null</td> </tr> </tbody> </table> <p>Какой или какие из запросов вернёт преподавателей, не преподающих ни одной дисциплины?</p> <ol style="list-style-type: none"> SELECT Teacher.* FROM Teacher LEFT JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T is null; SELECT Teacher.* FROM Teacher RIGHT JOIN discipline_teacher ON discipline_teacher.id_T = Teacher.id_T WHERE discipline_teacher.id_T is null; SELECT Teacher.* FROM Teacher JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T is null; SELECT Teacher.* FROM Teacher LEFT JOIN discipline_teacher ON Teacher.id_T= discipline_teacher.id_T WHERE discipline_teacher.id_T= null; | discipline | | | | id_dis | int | <pk> | not null | name | varchar(255) | | not null | english_name | varchar(255) | | null | short_name | varchar(24) | | not null | Teacher | | | | id_T | int | <pk> | not null | surname | varchar(25) | | not null | name | varchar(25) | | null | patronym | varchar(25) | | null | discipline_teacher | | | | id_dis_teach | int | <pk> | not null | id_T | int | <fk1> | null | id_dis | int | <fk2> | null | <p>ОПК-6.У.1</p> |
| discipline | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_dis | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| name | varchar(255) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| english_name | varchar(255) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| short_name | varchar(24) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Teacher | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_T | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| surname | varchar(25) | | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| name | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| patronym | varchar(25) | | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| discipline_teacher | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_dis_teach | int | <pk> | not null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_T | int | <fk1> | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| id_dis | int | <fk2> | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 27. | <p>Дана сводная таблица по работам студентов</p> <p>Работы</p> <table border="1"> <thead> <tr> <th>Студент</th> <th>Дисциплина</th> <th>Работа</th> <th>Балл</th> </tr> </thead> <tbody> <tr> <td>Иванов</td> <td>ООП</td> <td>1</td> <td>8</td> </tr> <tr> <td>Петров</td> <td>ООП</td> <td>2</td> <td>6</td> </tr> <tr> <td>Сидоров</td> <td>ПБД</td> <td>1</td> <td>4</td> </tr> <tr> <td>Иванов</td> <td>ПБД</td> <td>1</td> <td>Null</td> </tr> </tbody> </table> <p>Какое число вернет запрос ниже? SELECT avg(Балл) from Работы</p> <p>а. 6 б. 4,5 в. Null г. Ничего не выведет, выдаст ошибку, т.к. агрегатные функции не работают с null-значениями</p> | Студент | Дисциплина | Работа | Балл | Иванов | ООП | 1 | 8 | Петров | ООП | 2 | 6 | Сидоров | ПБД | 1 | 4 | Иванов | ПБД | 1 | Null | ОПК-6.У.1 | | | | | | | | | | |
|-----------------|---|-------------|------------|-------------|--------|------------|-----|-------------|------------|--------|----------|------------|---|---------|-----|-------------|--------|------------|-----|-------------|------------|-----------|-----------------|------------|----|------|-----|-------------|--------|------------|----|-----------|
| Студент | Дисциплина | Работа | Балл | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Иванов | ООП | 1 | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Петров | ООП | 2 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Сидоров | ПБД | 1 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Иванов | ПБД | 1 | Null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 28. | <p>Даны таблицы</p> <p>daily_temp</p> <table border="1"> <thead> <tr> <th>city</th> <th>day</th> <th>temperature</th> </tr> </thead> <tbody> <tr> <td>Москва</td> <td>14.05.2020</td> <td>15</td> </tr> <tr> <td>Симферополь</td> <td>15.05.2020</td> <td>23</td> </tr> <tr> <td>Мурманск</td> <td>16.05.2019</td> <td>5</td> </tr> </tbody> </table> <p>daily_temp_new</p> <table border="1"> <thead> <tr> <th>city</th> <th>day</th> <th>temperature</th> </tr> </thead> <tbody> <tr> <td>Москва</td> <td>14.05.2020</td> <td>15</td> </tr> <tr> <td>Симферополь</td> <td>14.05.2020</td> <td>23</td> </tr> <tr> <td>Санкт-Петербург</td> <td>15.05.2020</td> <td>15</td> </tr> </tbody> </table> <p>SELECT * FROM daily_temp</p> <p>ОПЕРАТОР</p> <p>SELECT * FROM daily_temp_new</p> <p>Какой оператор необходимо подставить вместо слова оператор, чтобы получить результат ниже?</p> <table border="1"> <thead> <tr> <th>city</th> <th>day</th> <th>temperature</th> </tr> </thead> <tbody> <tr> <td>Москва</td> <td>14.05.2020</td> <td>15</td> </tr> </tbody> </table> <p>INTERSECT</p> | city | day | temperature | Москва | 14.05.2020 | 15 | Симферополь | 15.05.2020 | 23 | Мурманск | 16.05.2019 | 5 | city | day | temperature | Москва | 14.05.2020 | 15 | Симферополь | 14.05.2020 | 23 | Санкт-Петербург | 15.05.2020 | 15 | city | day | temperature | Москва | 14.05.2020 | 15 | ОПК-6.У.1 |
| city | day | temperature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Москва | 14.05.2020 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Симферополь | 15.05.2020 | 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Мурманск | 16.05.2019 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| city | day | temperature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Москва | 14.05.2020 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Симферополь | 14.05.2020 | 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Санкт-Петербург | 15.05.2020 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| city | day | temperature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Москва | 14.05.2020 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 29. | <p>Дана предметная область , описывающая ведение дисциплин (discipline) преподавателями (teacher).</p> | ОПК-6.У.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



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

- a. **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));**
- б. 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%'));
- в. SELECT distinct Teacher.* FROM teacher JOIN discipline_teacher 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));

30.

Дана таблица студент

create table student

(

id_st int auto_increment primary key,

id_group int,

surname varchar(20),

ОПК-6.У.1


```
name          varchar(20) not null,
patronym      varchar(20),
);
```

Код которой/которых из процедур правильно добавляет в неё данные и возвращает идентификатор добавленного значения?

```
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 ;
```

a. +

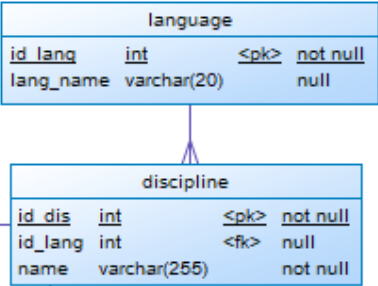
```
delimiter //
CREATE PROCEDURE ins_student (id_gr_ int,name_ varchar(15),
surname_ varchar(20),patronym_ varchar(25), out id_st_new int)
BEGIN
set id_st_new=LAST_INSERT_ID();
insert into student (id_st, surname, name,patronym,id_gr)
VALUES (id_st_new,surname_,name_,patronym_,id_gr_);

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;//
delimiter ;
```

в.

| | | |
|-----|--|-----------|
| | <pre> > 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;// delimiter ; г. </pre> | |
| 31. | <p>Даны таблицы языков преподавания и дисциплин</p>  <p>На таблицу дисциплин написан триггер before insert</p> <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. Никак</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> | ОПК-6.У.1 |

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

Таблица 19 – Перечень контрольных работ

| № п/п | Перечень контрольных работ |
|-------|---|
| 1 | Разработка концептуальной модели предметной области (6 семестр) |

| | |
|-----|---|
| 1.1 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. Вершины, название/текст которых содержит слово «конец», но не заканчивается им б. Вершины, у которых нет исходящих ребер в. Графы, в которых есть пара вершин, связанных ребрами в обе стороны г. Ширина графа в пикселях (от максимальной сумма координаты по горизонтали с шириной отнять минимальную левую координату) д. Пользователь- автор, графов с максимальных количеством вершин е. Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам ж. Вершина, нет входящих ребер от вершины 2 |
| 1.2 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. номера участков владельцев с отчеством, заканчивающимся на «вна» б. участки, на которых зарегистрировано более 1 постройки в. тип взносов, которые пока никто не оплатил г. Владелец (владельцы) участка максимальной площади д. Владельцы максимального количества участков е. Участки, на которых есть постройки всех типов ж. Участки, на которых нет бань |
| 1.3 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. квартиры владельцев, отчества которых заканчиваются на 'ич' б. владельцев, у которых есть квартиры на разных этажах |

| | |
|-----|--|
| | <ul style="list-style-type: none"> в. квартиры, в которых не записано ни одного показания счетчика газа г. владельцы квартир с минимальной жилой площадью д. этаж, на котором меньше всего квартир находится в собственности е. квартиры, в которых стоят все типы счетчиков ж. владелец, у которого нет трехкомнатных квартир |
| 1.4 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. аллеи, на которых встречаются разные виды кленов (клен в названии) б. аллеи, состоящие из разных пород деревьев в. породы, не высаженные в парке г. дерево, которое было посажено раньше всех д. порода, деревьев которой меньше всего е. аллея, на которой растут деревья всех пород, которые есть в парке ж. аллея, деревья на которой не высаживались в прошлом году |
| 1.5 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. аудитории, которых проходят экзамены по дисциплинам, имеющим в названии слова «базы данных» б. аудитории, где в один день проходит несколько экзаменов в. преподаватели, не принимающие экзаменов г. дисциплины самого последнего экзамена д. аудитории, в которых проходит больше всего экзаменов е. дисциплина, по которой есть экзамены у всех групп ж. преподаватель, не принимающий экзамены у группы Z2431 |
| 1.6 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках</p> |

| | |
|-----|--|
| | <p>диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. пациенты, приходившие на любые процедуры, связанные с электрофорезом б. пациент, приходивший к одному врачу и на прием и на процедуру в. пациенты, не ходившие в поликлинику г. процедуры с наименьшей стоимостью д. пациент, ходивший к наибольшему количеству врачей е. пациент, ходивший на все процедуры ж. пациент, не ходивший на процедуры к Иванову Ивану Ивановичу |
| 1.7 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. преподаватели, аз которыми закреплены дисциплины, начинающиеся со слова «автоматизирован» б. дисциплина, по которой есть и лабораторные и курсовая работа в. студенты, не прикрепившие ни одной работы г. студент, сдавший курсовую раньше всех д. студент, прикрепивший в этом месяце наибольшее число работ е. дисциплина, по которой есть все типы работ (КР, ЛР, практические) ж. преподаватель, которому не прикрепляли отчетов по курсовому проекту или работе |
| 1.8 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. спектакли, в которых используются костюмы, имеющие в названии слово шут б. костюм, в котором есть и куртка и штаны в. роль, на которую пока нет костюма г. автор, разработавший самый старый из костюмов д. спектаклю, к которому разработано наибольшее число костюмов е. костюм, в котором есть все типы деталей ж. автор, не разрабатывавший костюмы к «Золушке» |
| 1.9 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> |

| | |
|------|--|
| | <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все парковки, расположенные на линиях (не улицах или проспектах)(улица в адресе содержит «линия») б. владелец машины, у которого более одного места под машину в. машины, которые неизвестно когда уедут г. владелец машин, заезжавший раньше всех д. владелец машины, останавливавшийся на минимальном числе парковок е. машина, которая стояла на всех парковках Московского проспекта ж. владелец, не парковавшийся на Невском проспекте |
| 1.10 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. виды спорта, на которые есть безлимитные абонементы (со словом безлимит в названии типа) б. вид спорта, по которому занятия ведут различные тренеры в. вид спорта, по которому не куплено ни одного абонемента г. тренеры, ведущие занятия по виду спорта с минимальным число занятий в абонементе д. тренеры, которые ведут занятия по максимальному количеству видов спорта е. тренеры, не ведущие фитнес ж. абонемент, на все виды спорта, связанные с единоборствами, заканчивающимися на «до» |
| 1.11 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. коллекция обоев, начинающаяся со слова Элегия б. производитель, который производит как бумажные, так и флизелиновые обои в. текстура, обоев с которой нет |

| | |
|------|--|
| | <p>г. производитель с самой большой шириной рулона</p> <p>д. производитель обоев с максимальным количеством коллекций</p> <p>е. производитель, который производит обои из всех материалов</p> <p>ж. производитель, у которого нет обоев дороже 3000 рублей</p> |
| 1.12 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все маршруты, проходящие через остановку , в названии которой есть слово село (царское село, детское село) б. свежестроенная станция, к которой пока не ведет ни один маршрут в. маршрут, проходящий через Купчино и Шушары г. самая поздняя электричка, прибывающая на витебский вокзал д. маршруты с наибольшим количеством электричек е. маршрут, на котором нет электричек с режимом отправления по выходным. ж. маршрут, по которому ходят электрички со всеми режимами движения |
| 1.13 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. вакансии, имеющие в названии SQL б. подозрительные вакансии, где не прописаны обязанности в. работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора г. вакансия с наименьшей зарплатой д. вакансии с минимальным количеством обязательных требований е. требование, присутствующее во всех вакансиях на программиста ж. вакансии, в которых нет требования к опыту работы |
| 1.14 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или</p> |

| | |
|------|---|
| | <p>только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все расходы категорий, которые относятся к спорту (содержат слово спорт) б. месяц, в котором были разные статьи дохода в. статья доходов, по которой доходов не было г. категория, по которой наибольшие расходы в текущем году д. категория, по которой не было расходов в январе е. месяц, в котором были траты всех категорий расхода ж. месяц, в котором были траты максимального количества статей |
| 1.15 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. друзья, у которых номер телефона начинается на 8-911, б. Люди без меток, в. люди, которые относятся к соседям и коллегам одновременно г. самые старые люди среди контактов д. месяц, когда есть дни рождения у соседей, но нет у семьи е. метки, к которым относится максимальное количество людей ж. месяц, в котором есть дни рождения у людей со всеми метками |
| 1.16 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. направленности, в которых есть слово автоматизированный б. факультет, не принимающий ни на одно направление в. направление, на которое надо сдавать историю и обществознание г. факультет вуза, принимающий на наименьшее количество направлений д. город, в котором есть все направления е. вуз, с самым ранним началом приемной компании ж. направление, на которое не надо сдавать ЕГЭ по информатике |
| 1.17 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть</p> |

| | |
|------|--|
| | <p>поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. экскурсии в музеи (слово музей в любом месте названия) б. экскурсии, относящиеся к литературе и истории в. учитель, который не отвечает за экскурсии г. учащиеся, которые не ездили в музей истории религии д. экскурсия, собравшая наибольшее число участников е. самые дорогие экскурсии ж. учащийся, который был на всех экскурсиях |
| 1.18 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. породы, названия которых содержат слово «шпиц», но не начинается с него б. собаки без медалей в. собаки, у родителей которых один и тот же владелец г. владелец, у которого есть собаки всех пород д. владелец, у которого есть йоркширские терьеры, но нет мастиффов е. порода, собак которой меньше всего ж. владельцы самых старых собак |
| 1.19 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все посылки, отправляемые на улицу, в названии которой есть окончание «ая», но это не единственные буквы в нем б. улица, на которую никогда не доставляли посылки в. контактное лицо, получавшее посылки по разным адресам г. адреса, по которым доставили больше всего посылок д. посылка с самым маленьким весом е. контактное лицо, получавшее посылки от всех фирм ж. контактное лицо, никогда не получавшее посылок в январе |

| | |
|------|--|
| 1.20 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. достопримечательности, в которых есть слово «мать», но с него название не начинается б. город без достопримечательностей в. улица, на которой есть и памятники, и архитектурные комплексы г. город, в котором нет природных комплексов д. улица, на которой больше всего памятников е. города с самыми старыми достопримечательностями ж. улица, на которой есть достопримечательности всех типов |
| 1.21 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. станции, в названии которых есть слово «площадь», но оно на него не заканчивается б. линии без станций (в проекте) в. пересадочные станции с линии 1 на линию 2 г. линия с самым большим временем проезда д. линия, на которой собраны все станции, открываются раньше 7 утра е. станция с самым ранним открытием ж. линия на которой нет перегона больше 3 минут |
| 1.22 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все сотрудники, женского пола (отчество, оканчивается на «на») б. сотрудники, никогда не обращавшиеся в медпункт с жалобами в. сотрудники организации, делавшие за последний год прививки от столбняка и кори |

| | |
|------|--|
| | <p>г. сотрудник, последним сделавший флюорографию</p> <p>д. сотрудник, у которого сделаны все типы прививок</p> <p>е. сотрудник с самым большим количеством жалоб</p> <p>ж. сотрудник, не делавший прививки от энцефалита</p> |
| 1.23 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. любые помещения, где стоит компьютерная мебель (категория мебель и слово «комп») б. помещение, в котором нет никакого имущества в. материально ответственное лицо, у которого подотчетна и компьютерная техника и специальное оборудование г. имущество, списанное последним д. помещение, в котором есть имущество всех категорий е. мат. ответственное лицо, ответственное за наибольшее количество имущества ж. помещение, в котором нет компьютерной техники |
| 1.24 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M, M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. Все мероприятия, текущего года, в названии которых есть слово конференция, но с него с названия не начинаются б. Комната, в которой не проходят мероприятия в. Мероприятия, в которых участвуют сотрудники «подразделения 1» и «подразделения 2» г. Участники самых поздно заканчивающихся мероприятий д. Событие, с самым большим количеством пришедших на него участников.(участники на мероприятие могут быть приглашены и при этом могут придти/не придти). е. Мероприятие, в котором обязательно присутствие всех начальников отделов ж. Сотрудник, не посещавший мероприятий в июле прошлого года |
| 1.25 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть</p> |

| | |
|------|--|
| | <p>проставлена кратность(1:1, 1:М , М:М) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. Люди, выполняющие обязанности, имеющие в названии слово уборка, но не заканчивающиеся на него б. Обязанность, которую еще не распределили по выполняющим людям в. Обязанность, которую выполняет более одного человека г. Ежемесячная обязанность, которая выполняется первой по дате (дню) д. Категория, обязанности из которой выполняют все е. Категория, в которой меньше всего обязанностей ж. Человек, который не выполняет обязанности из категории «оплата счетов» |
| 1.26 | <p>Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.</p> <p>Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.</p> <p>На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:М , М:М) и степень (участвуют ли в связи все или только некоторые экземпляры сущности)</p> <p>База данных по теме «сбор в поездку» должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. Найти какие поездки за текущий год имели в названии слово «поход», но не заканчивались на него б. Найти какого типа поездок не совершали. в. Найти в какую поездку брались и очки, и купальные принадлежности г. Найти пользователя, у которого в текущем квартале больше 2 поездок д. Какие вещи берутся в поездки всех типов, куда ездил заданный пользователь е. Какие категории вещей не берутся в поездку типа «конференция» ж. Найти категорию с наибольшим количеством предметов |
| 2 | <p>Триггеры. Обеспечение активной целостности данных базы данных (7 семестр)</p> |
| 2.1 | <p>Реализовать для базы данных программы для рисования графов триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. Вершины, название/текст которых содержит слово «конец», но не заканчивается им б. Вершины, у которых нет исходящих ребер в. Графы, в которых есть пара вершин, связанных ребрами в обе стороны г. Ширина графа в пикселях (от максимальной сумма координаты по горизонтали с шириной отнять минимальную левую координату) д. Пользователь- автор, графов с максимальным количеством вершин е. Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам ж. Вершина, нет входящих ребер от вершины 2 |
| 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</p> |

| | |
|-----|--|
| | <p>можно добавить при необходимости.</p> <p>База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. номера участков владельцев с отчеством, заканчивающимся на «вна» б. участки, на которых зарегистрировано более 1 постройки в. тип взносов, которые пока никто не оплатил г. Владелец (владельцы) участка максимальной площади д. Владельцы максимального количества участков е. Участки, на которых есть постройки всех типов ж. Участки, на которых нет бань |
| 2.3 | <p>Реализовать для базы данных ТСЖ триггеры для всех событий (insert,delete, update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение. Вычисляемые поля можно добавить при необходимости.</p> <p>База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. квартиры владельцев, отчества которых заканчиваются на 'ич' б. владельцев, у которых есть квартиры на разных этажах в. квартиры, в которых не записано ни одного показания счетчика газа г. владельцы квартир с минимальной жилой площадью д. этаж, на котором меньше всего квартир находится в собственности е. квартиры, в которых стоят все типы счетчиков ж. владелец, у которого нет трехкомнатных квартир |
| 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"> а. аллеи, на которых встречаются разные виды кленов (клен в названии) б. аллеи, состоящие из разных пород деревьев в. породы, не высаженные в парке г. дерево, которое было посажено раньше всех д. порода, деревьев которой меньше всего е. аллея, на которой растут деревья всех пород, которые есть в парке ж. аллея, деревья на которой не высаживались в прошлом году |
| 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"> а. аудитории, которых проходят экзамены по дисциплинам, имеющим в названии слова «базы данных» б. аудитории, где в один день проходит несколько экзаменов в. преподаватели, не принимающие экзаменов г. дисциплины самого последнего экзамена д. аудитории, в которых проходит больше всего экзаменов е. дисциплина, по которой есть экзамены у всех групп ж. преподаватель, не принимающий экзамены у группы Z9431 |
| 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</p> |

| | |
|------|--|
| | <p>можно добавить при необходимости. База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. пациенты, приходившие на любые процедуры, связанные с электрофорезом б. пациент, приходивший к одному врачу и на прием и на процедуру в. пациенты, не ходившие в поликлинику г. процедуры с наименьшей стоимостью д. пациент, ходивший к наибольшему количеству врачей е. пациент, ходивший на все процедуры ж. пациент, не ходивший на процедуры к Иванову Ивану Ивановичу |
| 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. База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. преподаватели, аз которыми закреплены дисциплины, начинающиеся со слова «автоматизирован» б. дисциплина, по которой есть и лабораторные и курсовая работа в. студенты, не прикрепившие ни одной работы г. студент, сдавший курсовую раньше всех д. студент, прикрепивший в этом месяце наибольшее число работ е. дисциплина, по которой есть все типы работ (КР, ЛР, практические) ж. преподаватель, которому не прикрепляли отчетов по курсовому проекту или работе |
| 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. База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. спектакли, в которых используются костюмы, имеющие в названии слово шут б. костюм, в котором есть и куртка и штаны в. роль, на которую пока нет костюма г. автор, разработавший самый старый из костюмов д. спектаклю, к которому разработано наибольшее число костюмов е. костюм, в котором есть все типы деталей ж. автор, не разрабатывавший костюмы к «Золушке» |
| 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. База данных должна иметь структуру, позволяющую реализовать следующие запросы:</p> <ul style="list-style-type: none"> а. все парковки, расположенные на линиях (не улицах или проспектах)(улица в адресе содержит «линия») б. владелец машины, у которого более одного места под машину в. машины, которые неизвестно когда уедут г. владелец машин, заезжавший раньше всех д. владелец машины, останавливавшийся на минимальном числе парковок е. машина, которая стояла на всех парковках Московского проспекта ж. владелец, не парковавшийся на Невском проспекте |
| 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,</p> |

| | |
|------|---|
| | <p>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"> а. виды спорта, на которые есть безлимитные абонементы (со словом безлимит в названии типа) б. вид спорта, по которому занятия ведут различные тренеры в. вид спорта, по которому не куплено ни одного абонемента г. тренеры, ведущие занятия по виду спорта с минимальным числом занятий в абонементе д. тренеры, которые ведут занятия по максимальному количеству видов спорта е. тренеры, не ведущие фитнес ж. абонемент, на все виды спорта, связанные с единоборствами, заканчивающимися на «до» |
| 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 запросы:</p> <ul style="list-style-type: none"> а. коллекция обоев, начинающаяся со слова Элегия б. производитель, который производит как бумажные, так и флизелиновые обои в. текстура, обоев с которой нет г. производитель с самой большой шириной рулона д. производитель обоев с максимальным количеством коллекций е. производитель, который производит обои из всех материалов ж. производитель, у которого нет обоев дороже 3000 рублей |
| 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"> а. все маршруты, проходящие через остановку, в названии которой есть слово село (царское село, детское село) б. свежестроенная станция, к которой пока не ведет ни один маршрут в. маршрут, проходящий через Купчино и Шушары г. самая поздняя электричка, прибывающая на витебский вокзал д. маршруты с наибольшим количеством электричек е. маршрут, на котором нет электричек с режимом отправления по выходным. ж. маршрут, по которому ходят электрички со всеми режимами движения |
| 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"> а. вакансии, имеющие в названии SQL б. подозрительные вакансии, где не прописаны обязанности в. работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора г. вакансия с наименьшей зарплатой д. вакансии с минимальным количеством обязательных требований е. требование, присутствующее во всех вакансиях на программиста ж. вакансии, в которых нет требования к опыту работы |

| | |
|------|--|
| 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"> а. все расходы категорий, которые относятся к спорту (содержат слово спорт) б. месяц, в котором были разные статьи дохода в. статья доходов, по которой доходов не было г. категория, по которой наибольшие расходы в текущем году д. категория, по которой не было расходов в январе е. месяц, в котором были траты всех категорий расхода ж. месяц, в котором были траты максимального количества статей |
| 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"> а. друзья, у которых номер телефона начинается на 8-911, б. Люди без меток, в. люди, которые относятся к соседям и коллегам одновременно г. самые старые люди среди контактов д. месяц, когда есть дни рождения у соседей, но нет у семьи е. метки, к которым относится максимальное количество людей ж. месяц, в котором есть дни рождения у людей со всеми метками |
| 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"> а. направленности, в которых есть слово автоматизированный б. факультет, не принимающий ни на одно направление в. направление, на которое надо сдавать историю и обществознание г. факультет вуза, принимающий на наименьшее количество направлений д. город, в котором есть все направления е. вуз, с самым ранним началом приемной компании ж. направление, на которое не надо сдавать ЕГЭ по информатике |
| 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"> а. экскурсии в музеи (слово музей в любом месте названия) б. экскурсии, относящиеся к литературе и истории в. учитель, который не отвечает за экскурсии г. учащиеся, которые не ездили в музей истории религии д. экскурсия, собравшая наибольшее число участников е. самые дорогие экскурсии ж. учащийся, который был на всех экскурсиях |
| 2.18 | <p>Реализовать для базы данных собачьего питомника триггеры для всех событий</p> |

| | |
|------|---|
| | <p>(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"> породы, названия которых содержат слово «шпиц», но не начинается с него собаки без медалей собаки, у родителей которых один и тот же владелец владелец, у которого есть собаки всех пород владелец, у которого есть йоркширские терьеры, но нет мастиффов порода, собак которой меньше всего владельцы самых старых собак |
| 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 запросы:</p> <ol style="list-style-type: none"> все посылки, отправляемые на улицу, в названии которой есть окончание «ая», но это не единственные буквы в нем улица, на которую никогда не доставляли посылки контактное лицо, получавшее посылки по разным адресам адреса, по которым доставили больше всего посылок посылка с самым маленьким весом контактное лицо, получавшее посылки от всех фирм контактное лицо, никогда не получавшее посылку в январе |
| 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"> достопримечательности, в которых есть слово «мать», но с него название не начинается город без достопримечательностей улица, на которой есть и памятники, и архитектурные комплексы город, в котором нет природных комплексов улица, на которой больше всего памятников города с самыми старыми достопримечательностями улица, на которой есть достопримечательности всех типов |
| 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"> станции, в названии которых есть слово «площадь», но оно на него не заканчивается линии без станций (в проекте) пересадочные станции с линии 1 на линию 2 линия с самым большим временем проезда линия, на которой собраны все станции, открываются раньше 7 утра станция с самым ранним открытием линия на которой нет перегона больше 3 минут |
| 2.22 | <p>Реализовать для базы данных медпункта организации триггеры для всех</p> |

| | |
|------|--|
| | <p>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"> а. все сотрудники, женского пола (отчество, оканчивается на «на») б. сотрудники, никогда не обращавшиеся в медпункт с жалобами в. сотрудники организации, делавшие за последний год прививки от столбняка и кори г. сотрудник, последним сделавший флюорографию д. сотрудник, у которого сделаны все типы прививок е. сотрудник с самым большим количеством жалоб ж. сотрудник, не делавший прививки от энцефалита |
| 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 запросы:</p> <ul style="list-style-type: none"> а. любые помещения, где стоит компьютерная мебель (категория мебель и слово «комп») б. помещение, в котором нет никакого имущества в. материально ответственное лицо, у которого подотчетна и компьютерная техника и специальное оборудование г. имущество, списанное последним д. помещение, в котором есть имущество всех категорий е. мат. ответственное лицо, ответственное за наибольшее количество имущества ж. помещение, в котором нет компьютерной техники |
| 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"> а. Все мероприятия, текущего года, в названии которых есть слово конференция, но с него с названия не начинаются б. Комната, в которой не проходят мероприятия в. Мероприятия, в которых участвуют сотрудники «подразделения 1» и «подразделения 2» г. Участники самых поздно заканчивающихся мероприятий д. Событие, с самым большим количеством пришедших на него участников.(участники на мероприятие могут быть приглашены и при этом могут прийти/не прийти). е. Мероприятие, в котором обязательно присутствие всех начальников отделов ж. Сотрудник, не посещавший мероприятий в июле прошлого года |
| 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"> а. Люди, выполняющие обязанности, имеющие в названии слово уборка, но не заканчивающиеся на него б. Обязанность, которую еще не распределили по выполняющим людям в. Обязанность, которую выполняет более одного человека г. Ежемесячная обязанность, которая выполняется первой по дате (дню) д. Категория, обязанности из которой выполняют все |

| | |
|------|--|
| | <p>е. Категория, в которой меньше всего обязанностей</p> <p>ж. Человек, который не выполняет обязанности из категории «оплата счетов»</p> |
| 2.26 | <p>База данных по теме «сбор в поездку» должна иметь структуру, позволяющую реализовать следующие запросы:</p> <p>а. Найти какие поездки за текущий год имели в названии слово «поход», но не заканчивались на него</p> <p>б. Найти какого типа поездок не совершали.</p> <p>в. Найти в какую поездку брались и очки, и купальные принадлежности</p> <p>г. Найти пользователя, у которого в текущем квартале больше 2 поездок</p> <p>д. Какие вещи берутся в поездки всех типов, куда ездил заданный пользователь</p> <p>е. Какие категории вещей не берутся в поездку типа «конференция»</p> <p>ж. Найти категорию с наибольшим количеством предметов</p> <p>а.</p> |

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 обучающимся 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 также начальные сведения и указание источников для изучения основных возможностей языка SQL.

Структура предоставления лекционного материала:

- Раздел 1. Моделирование предметной области
- Тема 1.1 Концептуальное моделирование предметной области
- Тема 1.2. Логическое и физическое моделирование предметной области

- Тема 1.3 Нормализация и денормализация баз данных
 - Раздел 2. Введение в проектирование баз данных
- Тема 2.1. История развития и причины появления СУБД. Классификация СУБД.
- Тема 2.2. Модели данных.
- Тема 2.3. Место баз данных в программных системах. Базы данных в клиент-серверных приложениях и веб-приложениях. Встроенные базы данных для роботов и интернета вещей и мобильных приложений. Резидентные (in-memory) базы данных для обработки данных в режиме реального времени
 - Раздел 3. Реляционные базы данных
- Тема 3.1. Отношения и их свойства, ключи отношений, абстрактные операции манипулирования данными
- Тема 3.2. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных, манипулирования данными
- Тема 3.3. Реляционная алгебра
 - Раздел 4. Введение в язык SQL
- Тема 4.1. Оператор выборки в языке SQL, агрегатные функции. Применение поиска подстроки и регулярных выражений для обработки текстов как подготовка данных к машинному обучению. Трехзначная логика и обработка NULL-значений в языке SQL
- Тема 4.2. Запросы с подзапросами, экзистенциальные запросы, объединение, пересечение, разность запросов
- Тема 4.3. Использование представлений, управляющих конструкций в языке SQL
- Тема 4.4. Хранимые процедуры, триггеры. Назначение хранимых процедур.
- Тема 4.5. Триггеры. Назначение триггеров: обеспечение активной ссылочной целостности, управление основными данными
 - Раздел 5. Администрирование баз данных
- Тема 5.1 Управление доступом к базе данных. Управление правами доступа к базам данных.
- Тема 5.2 Индексация и оптимизация запросов.
- Тема 5.3 Целостность баз данных и параллельный доступ
- Тема 5.4. Выбор типа СУБД применительно к особенностям предметной области
 - Раздел 6. Объектные и объектно-реляционные базы данных
- Тема 6.1. Объектно-реляционные базы данных
- Тема 6.2. Объектные базы данных
 - Раздел 7. NoSQL базы данных
- Тема 7.1. Базы данных «Ключ-значение»
- Тема 7.2. Документные базы данных
- Тема 7.3. Графовые базы данных
- Тема 7.4. 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 <https://guap.ru/standart/doc> и титульным листом https://fs.guap.ru/docs/titul/2021/titul/lab_iodo.docx

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 пример использования оператором update и delete и merge

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

Задание для выполнения в лабораторной работе

- 1) Реализовать запросы а) .. в), указанные в варианте задания. .
(- Запрос на поиск по шаблону (поиск подстроки) (выполнить с единственным оператором like)

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

- Запрос на использование внешних соединений.)

Все запросы должны не содержать вложенных запросов или агрегатных функций

Лабораторная работа 5. Разработка SQL запросов: запросы с подзапросами

Задание для выполнения в лабораторной работе

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

(Запрос с условием на агрегатную функцию.

Запрос 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);

- 2) Создать пользователя, который обладает правами, указанными в варианте задания.

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

Задание для выполнения в лабораторной работе

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

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

Задание для выполнения в лабораторной работе

- 1)Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 9.
- 2)Сделать запрос выборки с условием к таблицам предку и потомку.
- 3)Придумать и создать пользовательский оператор для своей предметной области
- 4)Придумать и создать пользовательскую агрегатную функцию для своей предметной области

Лабораторная работа 9. Разработка документной базы данных

Задание для выполнения в лабораторной работе

- 1)Спроектировать структуру json файла , соответствующую предметной области по варианту задания.
- 2)Создать базу MongoDB с тестовыми данными для примера/

Лабораторная работа 10. Манипулирование данными в документной базе данных.

Задание для выполнения в лабораторной работе

- 1)Привести пример обновления и удаления данных из документной базы данных, разработанной в работе 11. (по 2 примера: один из вложенного документа, другой из документа коллекции верхнего уровня)
- 2)Выполнить запросы на выборку из базы данных в соответствии с вариантом задания .
(Запрос на поиск подстроки.

Запрос на использование одного документа несколько раз.

Запрос с условием на агрегатную функцию.

Запрос получением агрегатной функции от агрегатной функции.

Запрос на разность 2 запросов)

Распределение баллов и сроки сдачи лабораторных работ

Семестр 6

| № | Наименование лабораторной | Количество баллов |
|----|---|-------------------|
| 1. | Разработка физической модели базы данных с учетом декларативной ссылочной целостности | 5 |
| 2. | Создание и модификация базы данных и таблиц | 4 |
| 3. | Заполнение таблиц и модификация данных | 5 |
| 4. | Разработка SQL запросов: виды соединений и шаблоны | 7 |
| 5. | Разработка SQL запросов: запросы с подзапросами | 7 |
| 6. | Хранимые процедуры | 7 |
| | Контрольная работа «Разработка концептуальной модели предметной области» | 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 занятия 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 «Разработка концептуальной модели предметной области» 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

Спроектировать концептуальную модель предметной области (ER-диаграмму) в соответствии с вариантом задания.

Структура модели должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.

На диаграмме должно быть не менее 3 сущностей. Все сущности должны быть поименованы (уникально в рамках диаграммы) и иметь не менее одного атрибута и у него должен быть проставлен тип данных и задано уникальное в рамках диаграммы имя. Все связи должны быть поименованы, у них должна быть проставлена кратность(1:1, 1:M , M:M) и степень (участвуют ли в связи все или только некоторые экземпляры сущности).

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 |
|---|-----------------------------------|--------------------------------------|-----------------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |