МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Шуйский филиал ИвГУ

Кафедра математики, информатики и методики обучения

ОДОБРЕНО:

Руководитель ОП

_____ С.В. Белов
(подпись)

«21» февраля 2023 г.

Рабочая программа дисциплины

БАЗЫ ДАННЫХ

Уровень высшего образования:	бакалавриат
Квалификация выпускника:	бакалавр
Направление подготовки:	44.03.01 Педагогическое образование
Направленность (профиль) образовательной программы:	Информационные технологии и робототехника

Шуя

1. Цели освоения дисциплины

формирование системы знаний, умений и навыков, ориентированных на реализацию потребностей учителей в использовании баз данных для хранения, управления и анализа учебной информации и данных об обучающихся, а также осуществление практической подготовки обучающихся посредством выполнения определенных видов работ, связанных с будущей профессиональной деятельностью и направленных на формирование, закрепление, развитие практических навыков и компетенций по профилю образовательной программы.

2. Место дисциплины в структуре ОП

Дисциплина «Базы данных» является обязательной для изучения; относится к обязательной части блока дисциплин Б1, входит в предметно-методический модуль.

Успешное освоение данной дисциплины будет способствовать готовности обучающихся к освоению дисциплин: «Дистанционные технологии в образовательном процессе школы»; «Операционные системы»; «Web-технологии»; «Системное программирование»; «Прикладное программирование»; «Основы искусственного интеллекта»; «Логическое программирование»; успешному прохождению следующих практик: «Производственная практика, преподавательская практика (Предметно-методический модуль)», «Производственная практика, педагогическая (комплексная) практика (Предметно-методический модуль)», подготовке к сдаче и сдаче государственного экзамена, выполнению и защите выпускной квалификационной работы.

Студент, приступающий к изучению данной дисциплины, должен:

Знать:

– логические формы и процедуры;

– особенности системного и критического мышления;

– этапы развития предметной области;

Уметь:

– получать, обрабатывать и анализировать информацию из различных источников;

– совершенствовать свои профессиональные знания и умения на основе постоянного самообразования;

Владеть:

– навыками сопоставления разных источников информации с целью выявления их противоречий и поиска достоверных суждений;

– навыками определения практических последствий предложенного решения задач;

– умением выявлять и оценивать имеющиеся достоинства и недостатки профессиональной деятельности, делать обоснованные выводы.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

3.1. Компетенции, формированию которых способствует данная дисциплина

При реализации дисциплины формируются следующие компетенции в соответствии с ФГОС ВО по данному направлению подготовки:

а) универсальные компетенции (УК) и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции выпускника	Код и наименование индикатора достижения универсальной компетенции

Системное и критическое мышление	УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Демонстрирует знание особенностей системного и критического мышления и готовность к нему. УК-1.2. Применяет логические формы и процедуры, способен к рефлексии по поводу собственной и чужой мыслительной деятельности. УК-1.3. Анализирует источник информации с точки зрения временных и пространственных условий его возникновения. УК-1.4. Анализирует ранее сложившиеся в науке оценки информации. УК-1.5. Сопоставляет разные источники информации с целью выявления их противоречий и поиска достоверных суждений. УК-1.6. Аргументировано формирует собственное суждение и оценку информации, принимает обоснованное решение. УК-1.7. Определяет практические последствия предложенного решения задачи.

б) общепрофессиональные компетенции (ОПК) и индикаторы их достижения:

Наименование категории (группы) общепрофессиональных компетенций	Код и наименование общепрофессиональной компетенции выпускника	Код и наименование индикатора достижения общепрофессиональной компетенции
Научные основы педагогической деятельности	ОПК-8. Способен осуществлять педагогическую деятельность на основе специальных научных знаний	ОПК-8.1. Применяет методы анализа педагогической ситуации, профессиональной рефлексии на основе специальных научных знаний. ОПК-8.2. Проектирует и осуществляет учебно-воспитательный процесс с опорой на знания основных закономерностей возрастного развития когнитивной и личностной сфер обучающихся, научно-обоснованных закономерностей организации образовательного процесса, научных основ преподаваемых предметов, их истории и места в мировой культуре и науке. ОПК-8.3. Осуществляет мониторинг и отбор программ профессионального и личностного развития.

3.2. Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения формируемых компетенций

В результате освоения дисциплины обучающийся должен:

Знать:

– основы современных технологий сбора, обработки и представления информации (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– современные пакеты прикладных программ статистической обработки данных (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– научные основы преподаваемого предмета «Информационные технологии», его историю и место в мировой культуре и науке (ОПК-8 (индикаторы – 1, 2, 3);

– содержательные, методологические и мировоззренческие связи предметной области со смежными научными областями (ОПК-8 (индикаторы – 1, 2, 3);

Уметь:

– осуществлять корректный подбор методов анализа, проводить обработку данных исследования и правильную интерпретацию результатов (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– использовать современные информационно-коммуникационные технологии (включая пакеты прикладных программ, локальные и глобальные компьютерные сети) для сбора, обработки и анализа информации (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– читать и представлять статистические данные в различных видах (таблицы, диаграммы, графики) (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– руководствоваться в профессиональной деятельности научными основами преподаваемого предмета «Информационные технологии» (ОПК-8 (индикаторы – 1, 2, 3);

– соотносить основные этапы развития предметной области с ее актуальными задачами, методами и концептуальными подходами, тенденциями и перспективами ее современного развития (ОПК-8 (индикаторы – 1, 2, 3);

Владеть:

– средствами математического моделирования и анализа информации на компьютере с помощью электронных таблиц (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– научными основами преподаваемого предмета «Информационные технологии» (ОПК-8 (индикаторы – 1, 2, 3);

– теоретическими и практическими знаниями для постановки и решения исследовательских задач в предметной области и в области образования (ОПК-8 (индикаторы – 1, 2, 3).

4. Объём и содержание дисциплины

Очная форма обучения

Объём дисциплины составляет 3 зачётных единиц (81 час.)
Контактная работа 31,75 час.
Самостоятельная работа 49,25 час.
Практическая подготовка 34,5 час.

Виды учебных занятий и формы промежуточной аттестации	6 сем.	
	Контактная работа с преподавателем	Самостоятельная работа обучающегося
Лекции	10,5	49,25
Практические занятия	21	

Зачёт с оценкой	0,25	
Всего	31,75	49,25

4.1. Содержание дисциплины по разделам (темам), соотнесенное с видами и объемом учебных занятий

Очная форма обучения

№ п/п	Раздел (тема) дисциплины	Семестр	Виды учебных занятий, включая самостоятельную работу обучающихся и объем (в часах)				Формы текущего контроля успеваемости Формы промежуточной аттестации
			Занятия лекционного типа	Занятия семинарского типа		Самостоятельная работа обучающегося	
				Лаб.	Пр.		
1.	Сферы применения баз данных	6	1,5	-	1,5 ПП	4,5 ПП	практические работы
2.	Этапы проектирования баз данных	6	1,5	-	3 ПП	9,5	практические работы
3.	Реляционная модель данных	6	3	-	6 ПП	9 ПП	практические работы
4.	Язык SQL (Structured Query Language) и операции над данными	6	3	-	6 ПП	14	практические работы
5.	Системы управления базами данных	6	1,5	-	4,5 ПП	12,25	практические работы
ИТОГО			10,5	-	21	49,25	Зачёт с оценкой

4.2. Развернутое описание содержания учебного материала по разделам (темам)

Сферы применения баз данных. История развития автоматизированных систем обработки данных. Суть концепции баз данных. Развитие систем обработки данных. Современные тенденции в обработке данных. Принципы распределения обработки данных в архитектуре «клиент-сервер», классификация моделей «клиент-сервер» в архитектуре баз данных: файловый сервер, модель удаленного доступа, пассивный и активный сервера баз данных, трехзвенная архитектура и сервер приложений.

Этапы проектирования баз данных. Проектирование реляционных баз данных. Теория функциональных зависимостей и нормальных форм баз данных. Алгоритмы декомпозиции и синтеза при проектировании нормальных форм для реляционных баз данных. Современная трактовка реляционной базы данных: структурная целостность, уровень языков манипулирования данными в реляционных БД, ссылочная целостность.

Реляционная модель данных. Основные определения: домены, атрибуты, кортежи, отношения. Понятие отношения, домена, атрибута, кортежа, ранга (степени) отношения, схемы отношения. Понятие возможного ключа, первичного ключа. Определение тэта-сравнимых атрибутов. Инфологическое проектирование БД ER – диаграммы. Правила преобразования ER-диаграмм в реляционные таблицы. Схема отношения, его степень и

мощность. Правила преобразования модели «сущность-связь» в реляционные отношения. Разработка модели данных с использованием инструментального средства Oracle SQL Developer Data Modeler.

Язык SQL (Structured Query Language) и операции над данными. Операторы языка: CREATE, INSERT, UPDATE, DELETE, ALTER, SELECT. Использование SQL для выборки данных из таблицы: операторы в условиях AND, OR, IN, BETWEEN, LIKE, IS NULL. Запросы на выборку данных из нескольких таблиц. Соединения (JOIN). Вложенные запросы. Математические функции языка SQL. Функции для обработки строк. Функции для работы с датами. Работа с онлайн-тренажером sql-academy.org.

Системы управления базами данных. Базы данных MySQL. Управление базами данных при помощи phpMyAdmin. Взаимодействие с базами данных MySQL при помощи языка программирования Python. Создания баз данных SQLite с использованием языка программирования Python. Средство просмотра баз данных: SQLiteStudio. Создание баз данных в свободно-распространяемых средах: OpenOffice Base, LibreOffice Base и аналогичные. Базы данных PostgreSQL: установка, настройка, создание собственной базы данных, выполнение запросов, создание представлений.

5. Образовательные технологии, в том числе перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

В программе данного курса предусмотрено использование в учебном процессе активных форм проведения занятий и организации самостоятельной работы, рейтинговой технологии оценивания результатов обучения студентов. В учебном процессе используются как активные, так и интерактивные формы проведения занятий: технология учебной дискуссии, мозговой штурм.

Аудиторные занятия проводятся в интерактивной форме с использованием мультимедийного обеспечения (ноутбук, проектор) и технологии проблемного обучения.

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине: технологии смешанного обучения, мультимедиа технологии, облачные технологии, технологии визуализации (инфографика, презентации).

6. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся по дисциплине осуществляется в следующих формах:

- установка и настройка программного обеспечения для работы с базами данных;
- проектирование баз данных с использованием современных инструментальных средств, создание баз данных и их наполнение в системах управления базами данных;
- написание и тестирования запросов к базам данных на языке SQL (в том числе с применением онлайн-тренажеров);
- изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по базам данных;
- повторение материала с использованием конспектов.

Полностью весь методический материал по обеспечению самостоятельной работы студентов приводится в приложении 1 к настоящей программе. Доступ к методическим материалам осуществляется через сайт Шуйского филиала ИвГУ, в том числе через систему Moodle.

7. Характеристика оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Текущая аттестация осуществляется с использованием накопительного рейтингового оценивания.

Для текущего контроля успеваемости применяются:

- Практические работы на компьютере: 14 работ по 3 балла каждая (42 балла);
- Практические задания для самостоятельного выполнения: 24 задания по 1 баллу (24 балла);
- Выполнение проектного задания по предложенной теме (14 баллов);
- Опорный конспект по темам, предложенным преподавателем (10 баллов);

Помимо этого учитывается:

- посещаемость (отсутствие пропусков занятий) – 10 баллов.

Критерии оценки практических работ:

- 3 балла за практическую работу выставляется, если работа выполнена полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
- 2 балла выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
- 1 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
- 0 баллов выставляется, если работа не была выполнена.

Практические задания для самостоятельного выполнения оцениваются дихотомически:

- 1 балл за выполнение практического задания выставляется при условии его полного, самостоятельного и корректного выполнения, готовности студента ответить на вопросы преподавателя о проделанной работе;
- 0 баллов выставляется, если работа не была выполнена.

Критерии оценки проектного задания:

Максимально можно получить 14 баллов. За каждый скрипт можно получить от 0 до 2 баллов, в зависимости от правильности выполнения. Всего 5 * 2 = 10 баллов. От 0 до 4 баллов можно получить за полноту и корректность отчёта и выступления по проделанной работе.

Критерии оценки опорного конспекта по темам: Написание конспекта по каждой из пяти тем оценивается от 0 до 2 баллов: 2 балла – в конспекте отражены основные положения темы, имеются схемы, выделения особо значимой информации; 1 балл - в конспекте отражены основные положения темы, но отсутствуют признаки попытки переработки информации со стороны студенты; 0 баллов – основные положения темы не были отражены в конспекте.

Шкала количества баллов, выставляемых за посещение занятий:

Процент посещения занятий от их общего количества	Количество баллов, выставляемое за посещаемость
85%-100%	10 баллов
80%-84%	9 баллов
75%-79%	8 баллов
70%-74%	7 баллов
65%-69%	6 баллов

60%-64%	5 баллов
55%-59%	4 балла
50%-54%	3 балла
45%-49%	2 балла
40%-44%	1 балл
0%-39%	0 баллов

Оценка результатов освоения дисциплины при рейтинговой системе оценивания формируется по 100-балльной шкале из результатов текущего контроля успеваемости и результатов, полученных на зачёте. Зачёт с оценкой проводится в форме интегрального тестового задания, включающего вопросы по всем темам курса (максимально 30 баллов).

Оценка на зачёте выставляется по следующей шкале:

Шкала оценки результатов освоения дисциплины «Базы данных» при рейтинговой системе оценивания

Повышенный уровень	Базовый уровень	Пороговый уровень	Ниже порогового уровня
отлично	хорошо	удовлетворительно	неудовлетворительно
85-100 баллов	71-84 балла	61-70 баллов	60 и менее баллов

8. Учебно-методическое и информационное обеспечение дисциплины

Учебная и учебно-методическая литература:

– основная литература:

1. Гущин, А. Н. Базы данных : учебник : [16+] / А. Н. Гущин. – Москва : Директ-Медиа, 2014. – 266 с. : ил.,табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=222149. – ISBN 978-5-4458-5147-9. – DOI 10.23681/222149. – Текст : электронный.
2. Карпова, Т. С. Базы данных: модели, разработка, реализация : учебное пособие / Т. С. Карпова. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 241 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=429003. – Текст : электронный.
3. Основы построения баз данных : учебное пособие : [16+] / Д. В. Чмыхов, А. С. Сазонова, П. А. Тищенко [и др.]. – Москва ; Берлин : Директ-Медиа, 2021. – 124 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=602227. – Библиогр. в кн. – ISBN 978-5-4499-2428-5. – Текст : электронный.

– дополнительная литература:

1. Базы данных в высокопроизводительных информационных системах : учебное пособие / авт.-сост. Е. И. Николаев ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2016. – 163 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=466799. – Библиогр.: с. 161. – Текст : электронный.
2. Ермошин, А. В. Введение в базы данных : учебное пособие по курсу "Вычислительные сети и базы данных". Ч. 1 / А. В. Ермошин, В. В. Комиссарова ; Министерство образования и науки РФ; ГОУ ВПО "Шуйский государственный педагогический университет"; Кафедра информационных технологий и методики обучения. - Шуя : Изд-во ГОУ ВПО "ШГПУ", 2006. - 40 с.

3. Лазицкас, Е. А. Базы данных и системы управления базами данных : учебное пособие : [12+] / Е. А. Лазицкас, И. Н. Загумённикова, П. Г. Гилевский. – Минск : РИПО, 2016. – 267 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=463305. – Библиогр. в кн. – ISBN 978-985-503-558-0. – Текст : электронный.
4. Онопенко, Г. А. Базы данных : учебное пособие : [16+] / Г. А. Онопенко, Н. А. Вихорь ; Томский государственный архитектурно-строительный университет. – Томск : Томский государственный архитектурно-строительный университет (ТГАСУ), 2019. – 104 с. : схем, табл., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=694337. – ISBN 978-5-93057-908-6. – Текст : электронный.
5. Сидорова, Н. П. Базы данных: практикум по проектированию реляционных баз данных : учебное пособие : [16+] / Н. П. Сидорова ; Технологический университет, Институт техники и цифровых технологий, Факультет инфокоммуникационных систем и технологий. – Москва ; Берлин : Директ-Медиа, 2020. – 93 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=575080. – Библиогр.: с. 85. – ISBN 978-5-4499-0799-8. – Текст : электронный.
6. Сирант, О. В. Работа с базами данных : курс лекций / О. В. Сирант, Т. А. Коваленко. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 150 с. : схем., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=428978. – Библиогр. в кн. – Текст : электронный.
7. Чурбанова, О. В. Базы данных и знаний. Проектирование баз данных в Microsoft Access : учебно-методическое пособие / О. В. Чурбанова, А. Л. Чурбанов ; Северный (Арктический) федеральный университет им. М. В. Ломоносова. – Архангельск : Северный (Арктический) федеральный университет (САФУ), 2015. – 152 с. : ил., схем., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=436230. – Библиогр. в кн. – ISBN 978-5-261-01029-6. – Текст : электронный.
8. Щелоков, С. А. Базы данных : учебное пособие / С. А. Щелоков ; Оренбургский государственный университет, Кафедра программного обеспечения вычислительной техники и автоматизированных систем. – Оренбург : Оренбургский государственный университет, 2014. – 298 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=260752. – Библиогр. в кн. – Текст : электронный.

Интернет-ресурсы:

1. Интерактивный онлайн курс по SQL. Упражнения и задачи по написанию SQL запросов в MySQL. Основы для начинающих. Режим доступа: https://sql-academy.org/
2. SQL OnLine IDE. Режим доступа: https://sqliteonline.com/
3. Курс «SQL для непрограммистов». Вебурситет. Режим доступа: https://www.webursitet.ru/product/sql-dlya-neprogrammistov.html
4. Презентации к учебнику информатики К.Ю. Полякова и Е.А. Еремина. 10 и 11 классы. ФГОС. Углублённый уровень. Режим доступа: https://kpolyakov.spb.ru/school/probook/slides.htm
5. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/
6. Образовательные сайты и порталы. http://school2100.com/

7. Базы данных на уроках информатики. НОУ Интуит. https://intuit.ru/studies/professional_skill_improvements/1327/info
8. Профессиональный сайт учителей «Методисты». http://nsportal.ru/

Современные профессиональные базы данных и информационно-справочные системы:
ЭБС «Университетская библиотека онлайн»;
Коллекция Шуйского филиала ИвГУ на национальном цифровом ресурсе «Руконт».
Консультант Плюс http://www.consultant.ru/
Методическое хранилище Ивановского института развития образования].-http://www.iroio.ru/;

Программное обеспечение:
В процессе изучения дисциплины используются офисные пакеты Microsoft Office, LibreOffice, инструментальное средство Oracle SQL Developer Data Modeler, программная среда для веб-разработчиков Open Server Panel, программа для просмотра и чтения файлов PDF Adobe Acrobat Reader DC, программа для воспроизведения флэш-анимации в браузерах Adobe Flash Player, вэб-обозреватели Google Chrome, Opera, Антивирус Касперского, 1С университет.

9. Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения занятий лекционного типа, укомплектованная специализированной мебелью и техническими средствами обучения, служащими для предоставления учебной информации большой аудитории.
2. Учебная аудитория для проведения занятий семинарского типа, укомплектованная специализированной мебелью и техническими средствами обучения.
3. Учебная аудитория для проведения групповых и индивидуальных консультаций, укомплектованная специализированной мебелью и техническими средствами обучения.
4. Учебная аудитория для проведения текущего контроля и промежуточной аттестации, укомплектованная специализированной мебелью и техническими средствами обучения.
5. Помещение для самостоятельной работы, оснащенное специализированной мебелью, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.
6. Учебно-наглядные пособия для занятий лекционного типа, обеспечивающие тематические иллюстрации: презентации по базам данных и системам управления базами данных.

Авторы рабочей программы дисциплины: А.В. Ермошин, доцент кафедры математики, информатики и методики обучения, кандидат технических наук, доцент; В.А. Смирнов, преподаватель (аспирант) кафедры математики, информатики и методики обучения.

Программа рассмотрена на заседании кафедры математики, информатики и методики обучения «16» февраля 2023 г., протокол № 7

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

<u>Приложение 1.</u> Методические указания для обучающихся по освоению дисциплины

Сопровождение самостоятельной работы студентов по курсу «Базы данных» может быть организовано в следующих формах:

– согласование индивидуальных заданий (виды и темы заданий, сроки представления результатов) самостоятельной работы студента в пределах часов, отведенных на самостоятельную работу;

– консультации (индивидуальные и групповые), в том числе с применением дистанционных технологий взаимодействия;

– промежуточный контроль хода выполнения заданий, строящихся на основе различных способов самостоятельной деятельности в открытой информационной среде.

Преподаватель определяет содержание самостоятельной работы, график её выполнения; создает сетевую информационную и коммуникационную среду для организации самостоятельной работы.

Одной из важнейших особенностей заданий для самостоятельной работы по курсу «Базы данных» является их ориентация на мотивирование самообразовательной деятельности с использованием ИКТ.

К типовым заданиям для самостоятельной работы студентов относятся:

– Конспектирование и реферирование литературных источников по тематике учебного курса.

– Решение задач, связанных с установкой программного обеспечения для работы с базами данных, написание запросов на языке SQL для решения прикладных задач;

– Выполнение индивидуального проектного задания, связанного с созданием базы данных по выбранной предметной области, наполнением её данными и написанием запросов к ней для обработки этих данных;

– Изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по базам данных;

– Повторение материала с использованием конспектов.

Перечень вопросов для самостоятельного изучения

1. Дайте определение реляционной базы данных? Роль системы управления базами данными (СУБД) в организации.
2. Архитектура современных СУБД. В чем суть логического и инфологического моделирования?
3. Этапы проектирования СУБД. Каковы принципы построения ER-модели?
4. Иерархическая модель данных. Основные понятия, графическое изображение, приме-ры.
5. Сетевая модель данных. Основные понятия, графическое изображение, примеры.
6. Реляционная модель данных. Основные понятия, графическое изображение, примеры.
7. Нормализация таблиц реляционной базы данных. Определение, виды, понятия.
8. Первая нормальная форма реляционной модели данных. Определение, требования, примеры.
9. Вторая нормальная форма реляционной модели данных. Определение, требования, примеры.
10. Третья нормальная форма реляционной модели данных. Определение, требования, примеры.

11. Какие задачи по проектированию БД могут решаться с помощью инструментальных средств ErWin/BpWin?
12. Каково назначение языков DDL и DML. Как создать таблицу БД используя DDL.
13. Какие существуют операторы языка SQL для чтения и записи данных?
14. Какие существуют операторы языка SQL для обновления (редактирования) и удале-ния данных?
15. Какие существуют операторы языка SQL предоставления прав доступа к отдельным объектам БД различным пользователям?
16. В чем отличия архитектуры клиент-сервер от архитектуры файл-сервер? Каковы осо-бенности работы БД в многопользовательском режиме?
17. Как осуществить доступ к БД из приложений под Windows, используя VBA и ADO.
18. Опишите основные этапы проектирования приложений БД под WEB. Как осуще-ствить передачу данных с WEB-страницы серверу БД?
19. Перечислите функции администратора БД. Как осуществить физический пере-нос объектов БД с данными между различными серверам (например, в MS SQL Server).
20. Перечислите основные инструментальные средства СУБД MS SQL Server?
21. Перечислите основные инструментальные средства СУБД Oracle?
22. Какие функции управления данными могут быть реализованы с помощью Oracle enterprise manager?
23. Из каких ключевых блоков состоит программа PL/SQL?
24. Как осуществить запрос нескольких строк из базы данных, используя курсоры PL/SQL?
25. Физические модели данных. Определения, назначение информационной модели, цели.
26. Правила составления условий отбора данных. Операторы сравнения, использование функций.

Задания для организации практических занятий:

Практическая работа №1 «Выделение таблиц предметной области»

Одним из основных моментов при создании реляционной базы данных является процесс выделения таблиц, которые должны использоваться для хранения данных конкретной предметной области.

Например, в информационной системе «Расписание занятий» должны присутствовать следующие таблицы:
- Таблица аудиторий (код аудитории, название, расположение в пределах здания);
- Таблица преподавателей (табельный номер, фамилия, имя, отчество, должность);
- Таблица дисциплин (код дисциплины, название дисциплины);
- Таблица расписания звонков (номер пары, время начала занятия, время конца занятия);
- Таблица типов занятий (номер типа, название (лекция, лабораторная и т.д.));
- Таблица расписания (номер пары, код аудитории, табельный номер преподавателя, код дисциплины, номер пары, номер типа занятия, дата проведения).

Упражнение 1. Укажите данные, которые необходимо хранить при организации:

а) *системы регистрации учащихся*.

Подробное описание: Школьному округу ABC требуется информационная онлайн-система для регистрации учащихся и сбора информации об учащихся. Эта система должна быть реализована как интерактивный процесс, чтобы все новые учащиеся могли регистрироваться онлайн. Кроме того, у существующих учащихся должна быть возможность обновлять и просматривать всю информацию. Создайте список важных данных, которые необходимо собрать и сохранить в базе данных регистрации учащихся.

б) *системы управления библиотекой*.

Подробное описание: Сообщество XYZ планирует создать систему управления библиотекой. База данных должна обрабатывать все операции библиотеки. В базе данных должны храниться все данные, относящиеся к управлению библиотечным фондом, обслуживанию посетителей и повседневной работе библиотеки. Создайте список важных данных, которые необходимо собрать и сохранить в базе данных управления библиотекой.

Упражнение 2. Определите возможные таблицы и поля в них, опираясь на предложенные сценарии.

a) Book.com — интернет-магазин, где покупатели могут посмотреть каталог и выбрать интересующие их товары.

a. У каждой книги есть название, номер ISBN, год издания и цена. В базе также хранятся сведения об авторах и издателях книг.

b. Сведения об авторах включают в себя имя, адрес и ссылку на главную страницу сайта.

c. Сведения об издателях включают в себя имя, адрес, номер телефона и ссылку на главную страницу сайта.

d. В базу внесены данные нескольких складов, включая код, адрес и номер телефона.

e. На складе хранится некоторое количество книг. Одна и та же книга может храниться на нескольких складах.

f. В базе данных регистрируется число экземпляров книги, хранящихся на различных складах.

g. Книжный магазин хранит данные покупателей: имя, адрес, адрес электронной почты и номер телефона.

h. У покупателя имеется несколько корзин. Корзине присваивается идентификатор Shopping_Cart_ID, и она содержит несколько книг.

i. Некоторые корзины могут содержать несколько экземпляров одной и той же книги. В базе данных регистрируется число экземпляров каждой книги, находящихся в любой корзине.

j. На данном этапе для выполнения транзакции требуется дополнительная информация. Как правило, покупателю нужно указать или выбрать платежный адрес, адрес доставки, способ доставки и платежные данные, например номер кредитной карты. После размещения заказа покупатель получает уведомление по электронной почте.

б) Компания ABC Ltd планирует компьютеризировать свою систему продаж и управления запасами. Проведенный анализ целесообразности показал, что этой компании настоятельно рекомендуется внедрить реляционную базу данных. Система продаж и управления запасами ABC функционирует следующим образом:

a. Покупатели отправляют заказы на приобретение товаров. Каждый заказ может содержать несколько запросов на различное количество одного или нескольких товаров из ассортимента ABC. ABC ведет складской учет. Складская ведомость содержит информацию о каждом товаре: описание, сведения о предпочтительном поставщике, количество единиц товара в наличии, возможность повторного заказа и прочие данные.

b. Получив заказ от покупателя, ABC доставляет имеющиеся в наличии товары и создает счет на отправленные покупателю товары. Товары, которых нет в наличии, помещаются в список отложенных заказов и, как правило, повторно заказываются у предпочтительного поставщика. В некоторых случаях товары заказываются у других поставщиков.

c. Получив счета от компании ABC, покупатели осуществляют оплату. При этом покупатель может оплатить один счет, некоторые позиции счета или несколько счетов (полностью или частично).

Упражнение 3. Вам предстоит попытаться представить выбранную предметную область в виде набора таблиц. В зависимости от собственных интересов выберите один из следующих вариантов:

- Система хранения тем выпускных квалификационных работ (по окончании обучения студент выполняет дипломную работу, руководителем которой является единственный преподаватель, но при этом преподаватель может руководить несколькими студентами);
- Система хранения данных о студентах (студенты распределены по группам и направлениям подготовки, каждая из групп прикреплена к конкретному факультету);
- Система хранения данных о пациентах в медицинской организации (пациент может посещать различных врачей, каждый из которых может иметь различную специализацию. Система хранит дату посещения специалиста, а также сканированную версию медицинского заключения, но не предполагает поиска и систематизации заболеваний);
- Гостиничный комплекс (жильцы заселяются только в одноместные номера, имеющие разную стоимость. Система хранит дату заселения и дату выселения. Кроме того, в каждом номере есть определенное оборудование: телевизор, холодильник и т.д. Хранить детальную информацию (марку, производителя) об оборудовании необходимости нет, однако важно учитывать, что в номере может быть несколько единиц каждого вида оборудования).

Практическая работа №2 «Создание реляционной модели базы данных»

Описание ситуации: имеем базу данных, состоящую из следующих таблиц:

Сотрудники				
Табельный номер	**ФИО**	**З/П**	**Номер должности**	**Номер отдела**
1	Федоров Петр Иванович	15000	2	2
2	Сидоров Кирилл Игоревич	20000	3	2
3	Петров Владимир Альбертович	18000	2	2

4	Колотилов Иван Владимирович	23000	1	1
...	...	...	...	...

Должности	
Номер должности	**Название должности**
1	Бухгалтер
2	Программист
3	Администратор баз данных
...	...

Отделы	
Номер отдела	**Название отдела**
1	Бухгалтерия
2	IT-отдел
...	...

В этих таблицах присутствуют следующие поля:

Должности		
Номер должности	Целое число	Первичный ключ
Название должности	Строка	

Отделы		
Номер отдела	Целое число	Первичный ключ
Название отдела	Строка	

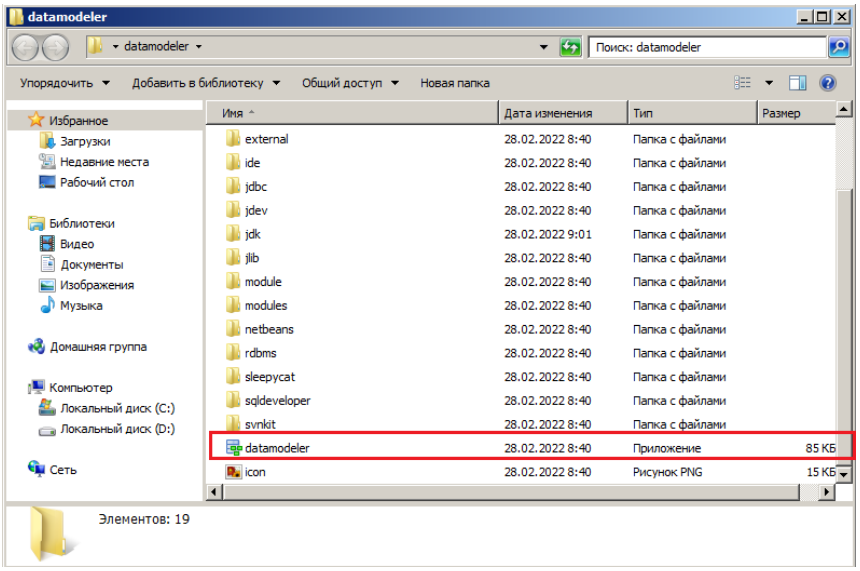
Сотрудники		
Табельный номер	Целое число	Первичный ключ
ФИО	Строка	
З/П	Целое число	
Номер должности	Целое число	Внешний ключ (таблица «Должности»)
Номер отдела	Целое число	Внешний ключ (таблица «Отделы»)

Информация о данных таблицах может быть визуализирована с помощью диаграммы «сущность-связь» (ERD).

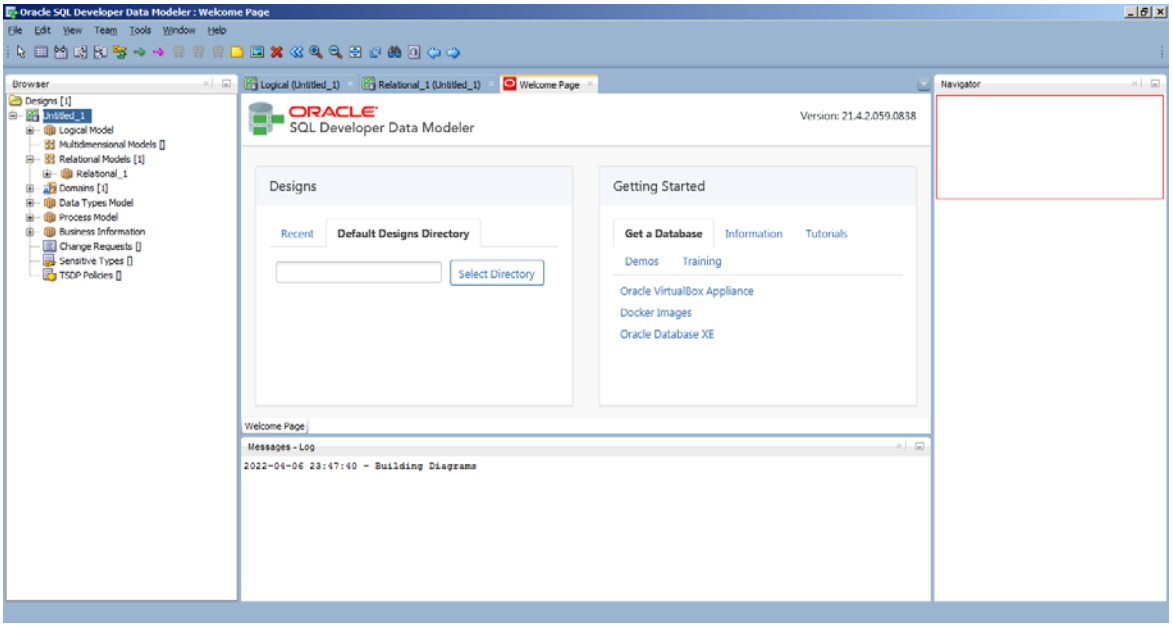
Цель занятия – построение реляционной модели базы данных в Oracle SQL Developer Data Modeler.

Данная программа является бесплатной. Ранее она могла быть загружена с официального сайта Oracle. В данный момент рекомендую воспользоваться ссылкой для скачивания: https://www.softpedia.com/get/Internet/Servers/Database-Utils/Oracle-SQL-Developer-Data-Modeler.shtml

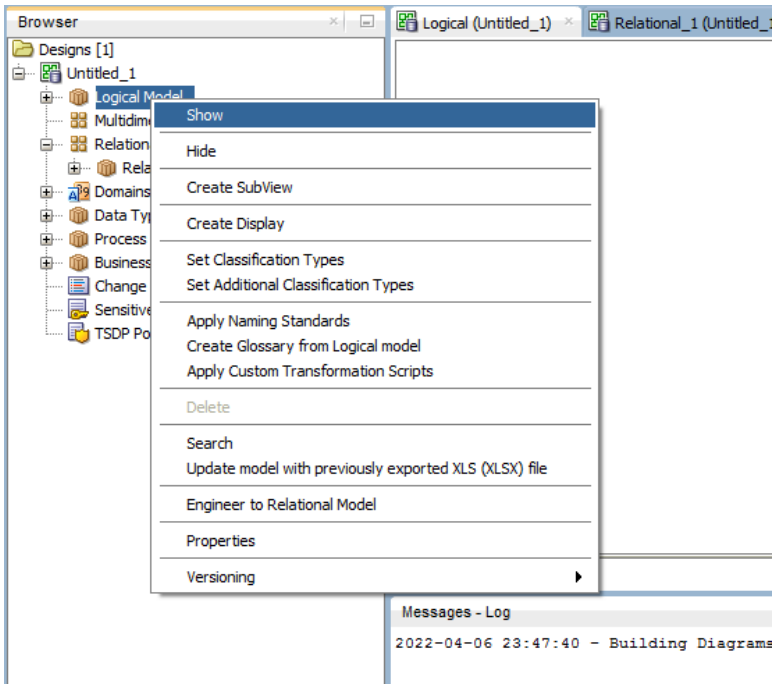
Скачанный файл представляет собой zip-архив, который необходимо распаковать. Из архива будет извлечена папка со следующим содержимым:



Файл, который запускает среду разработки, выделен красным прямоугольником.
Главное окно программы при запуске:

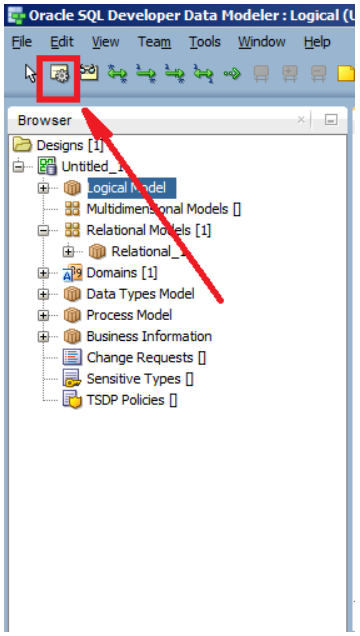


Отсюда переходим во вкладку Logical (логическая модель базы данных).
Если вкладки нет, то её можно открыть с помощью контекстного меню пункта «Logical Model»:

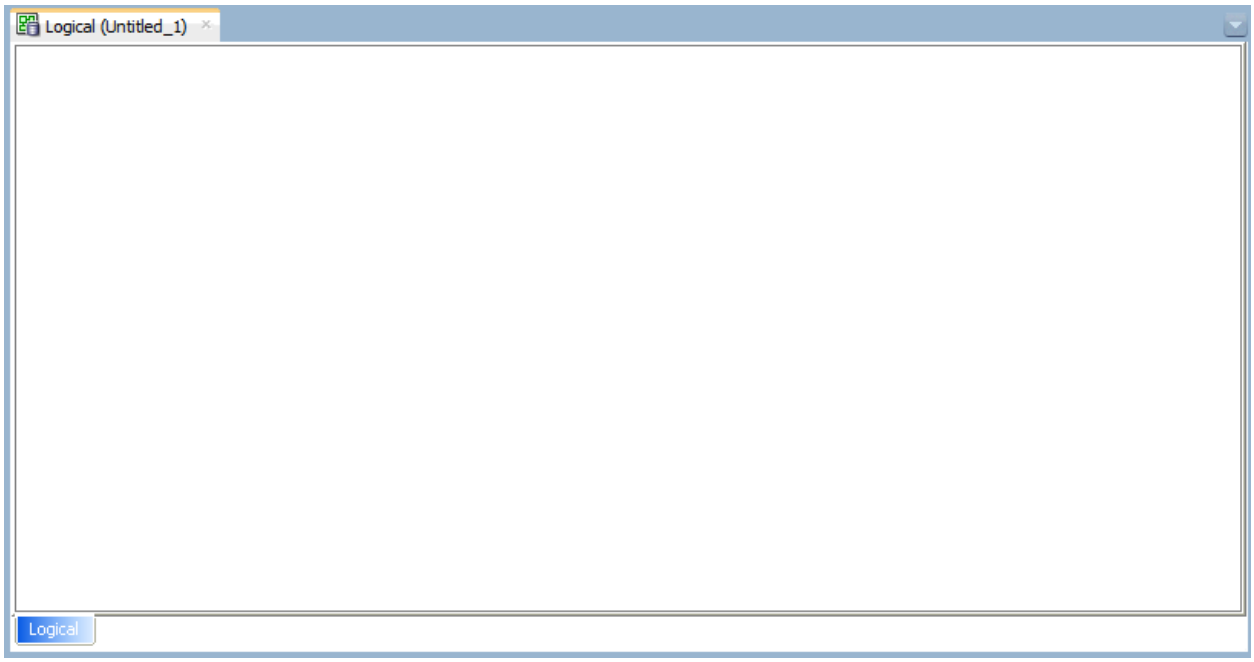


Таблицам в реляционной модели соответствуют сущности в логической модели. Столбцам таблицы – атрибуты сущности.

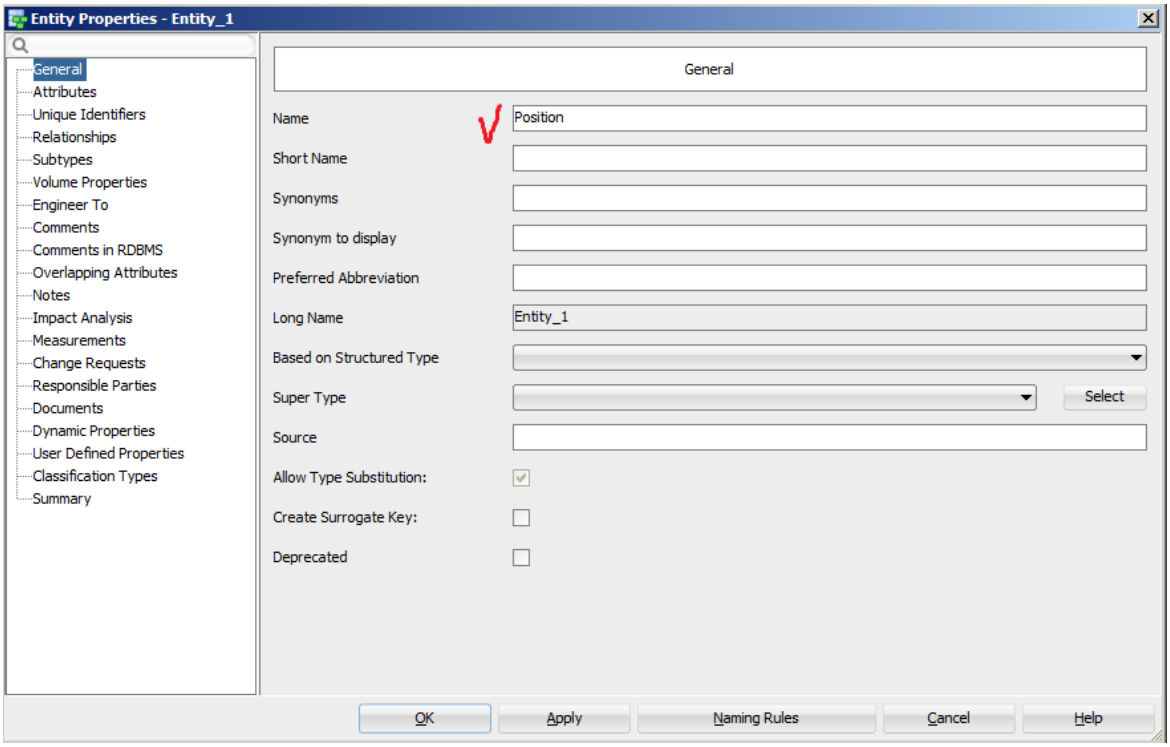
Добавление «сущности» в модель производится при помощи кнопки «New Entity»:



После выбора инструмента курсор в центральной панели (на рис. ниже) сменит изображение на значок «выстрела». На этой панели можно нажать в любое свободное место.

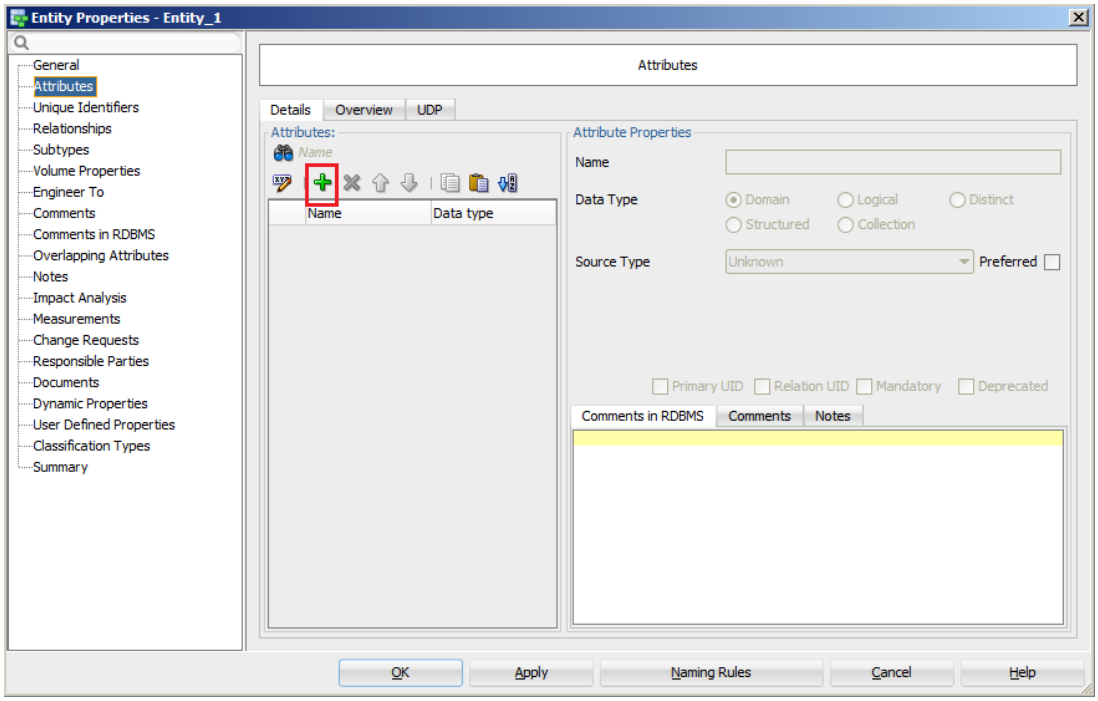


В появившемся окне указываем имя сущности:

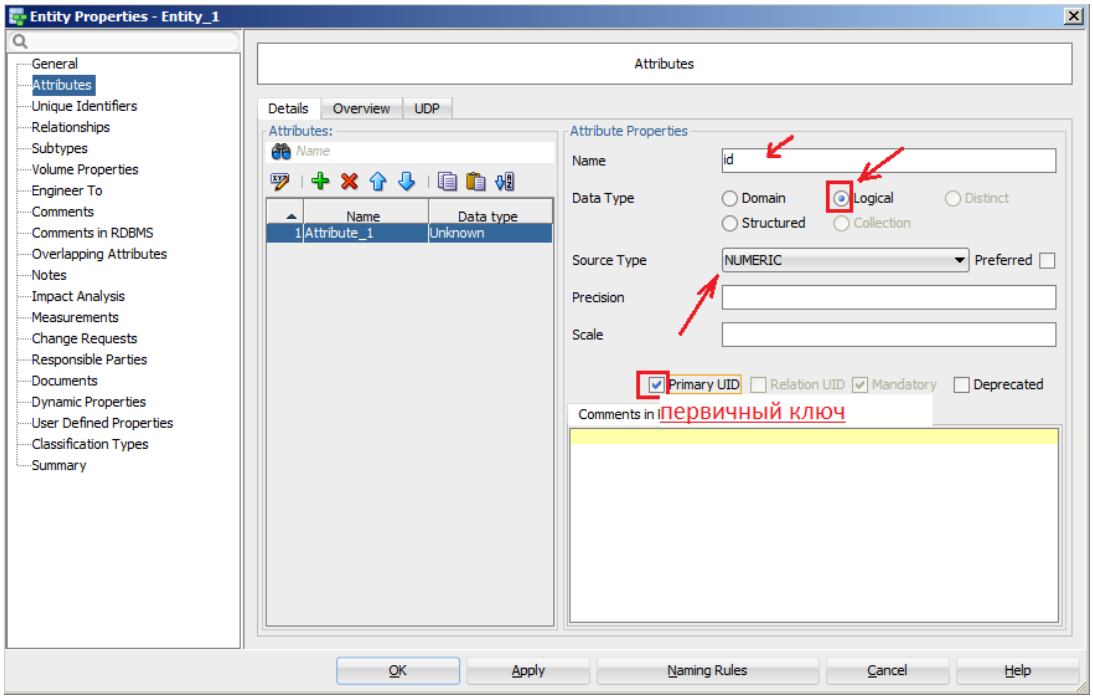


После чего переходим на вкладку «Attributes»:

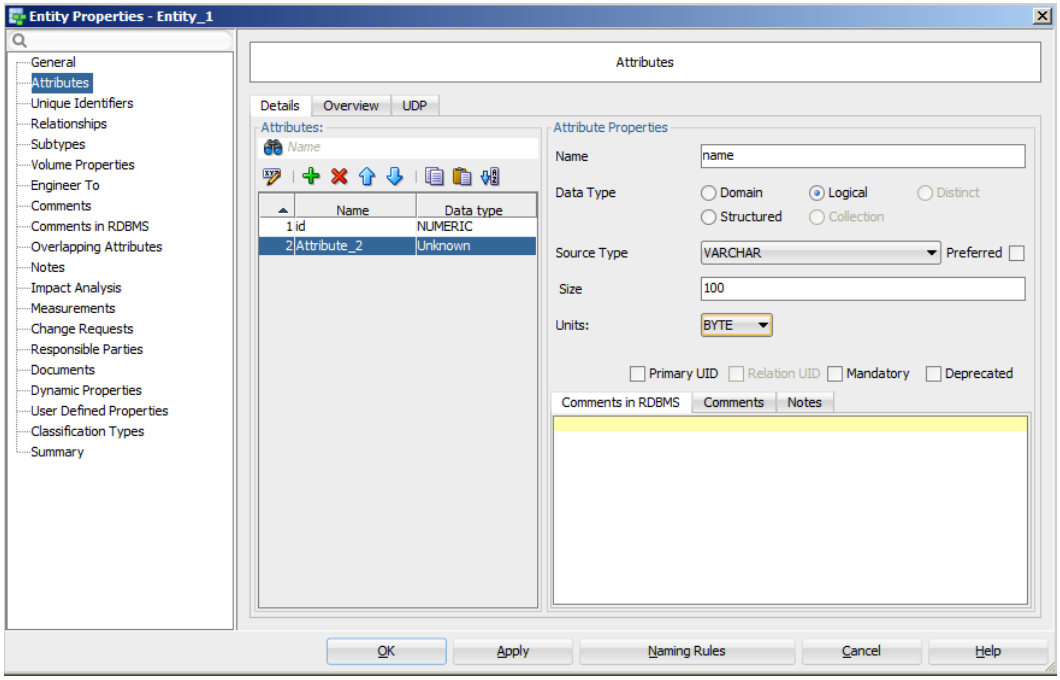
Кнопкой со знаком «+» добавляем новый атрибут:



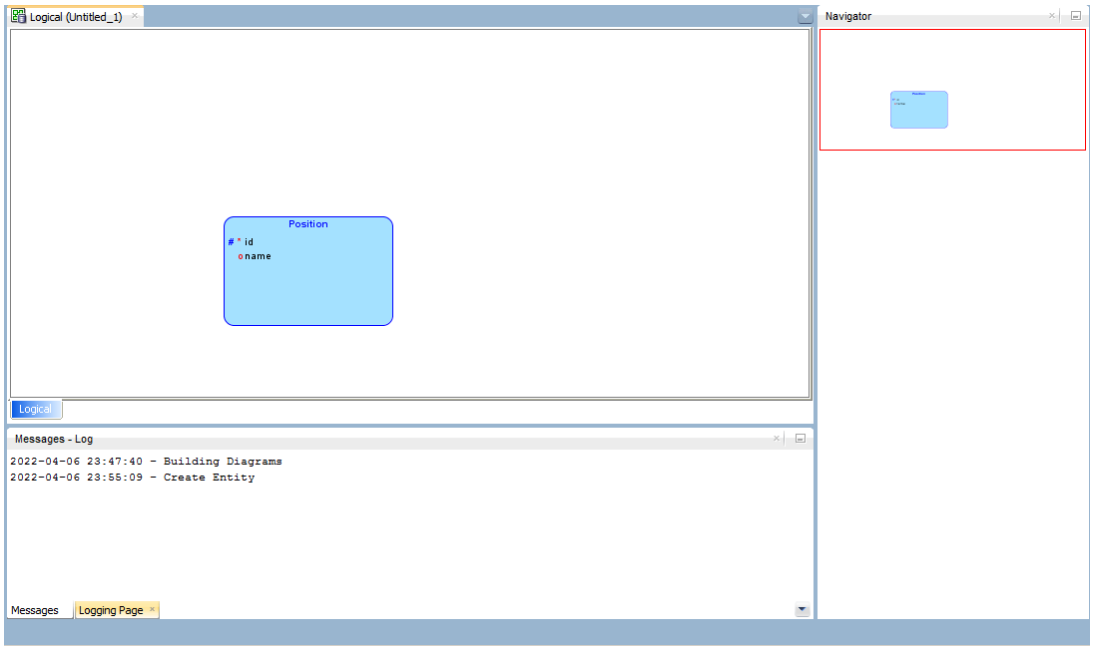
Указываем информацию о добавляемом атрибуте (номер сотрудника – числовой атрибут, будет являться первичным ключом в таблице):



Добавляем кнопкой со знаком «+» еще один атрибут (название должности, строка, состоящая из не более чем 100 символов):

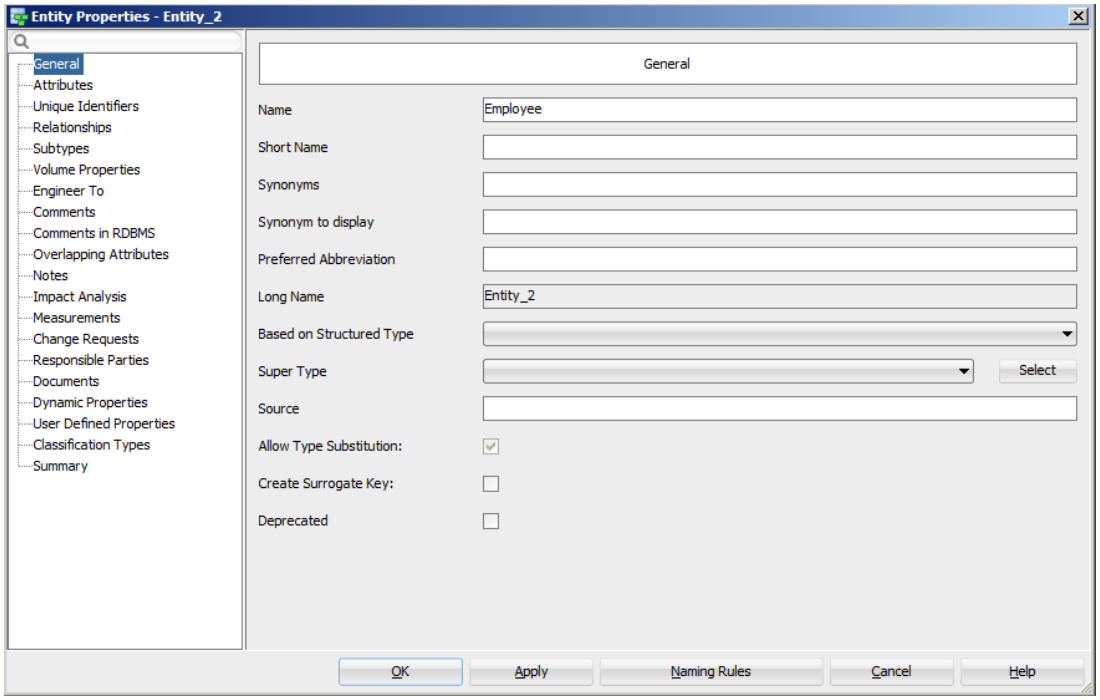


После нажатия «ОК»:

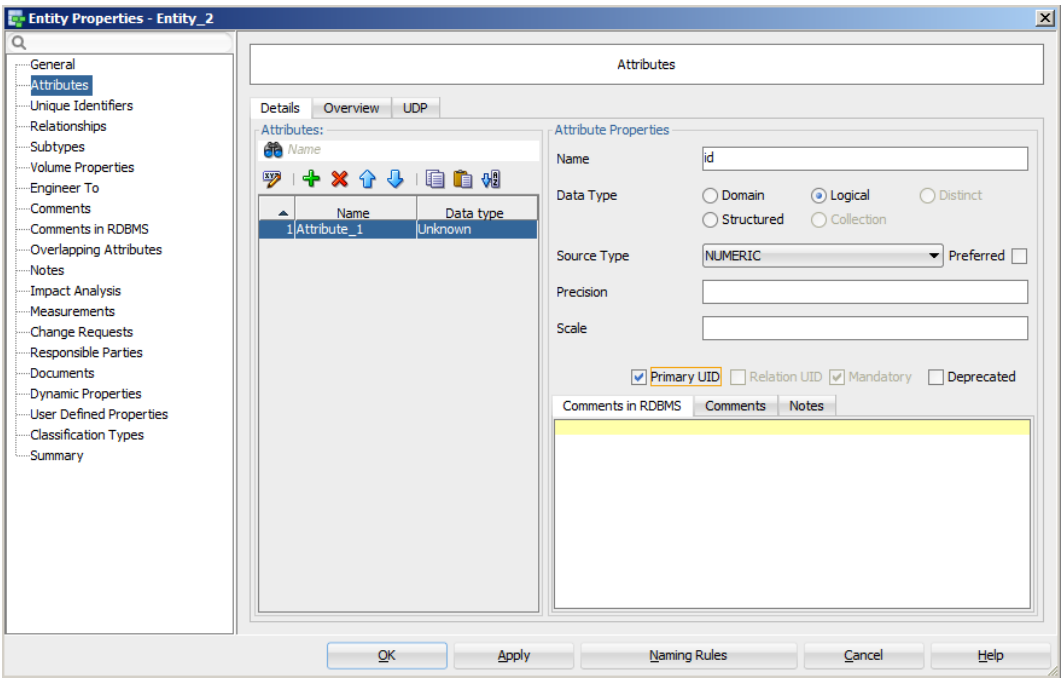


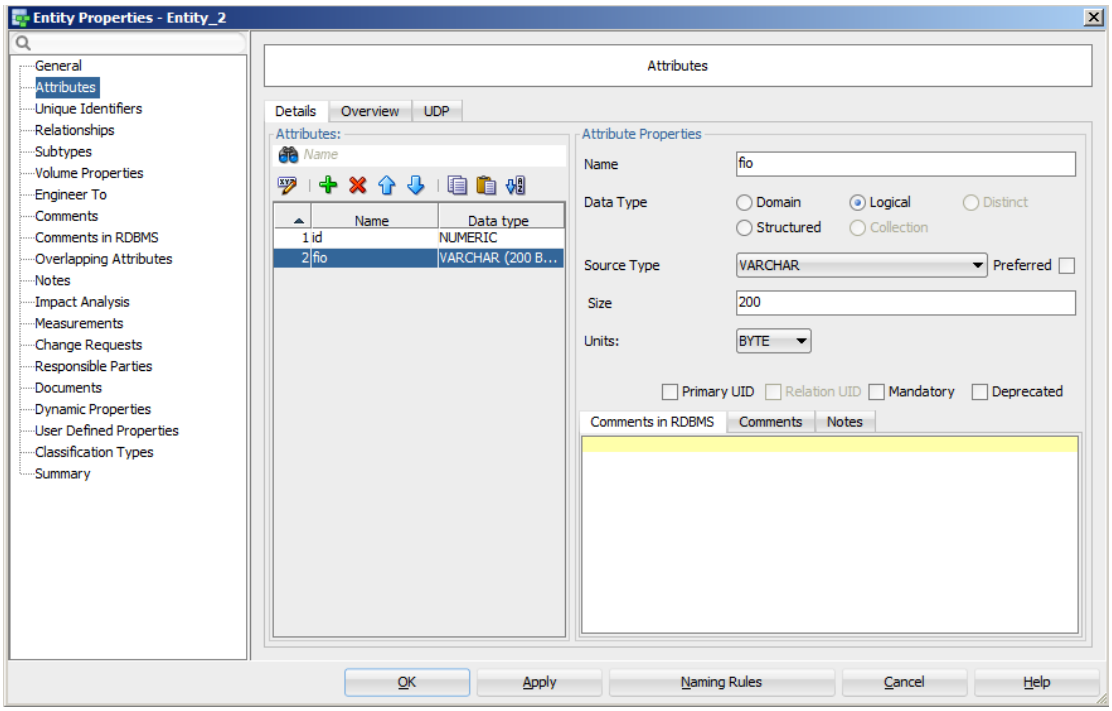
Сущность «Отдел» добавляется аналогично.

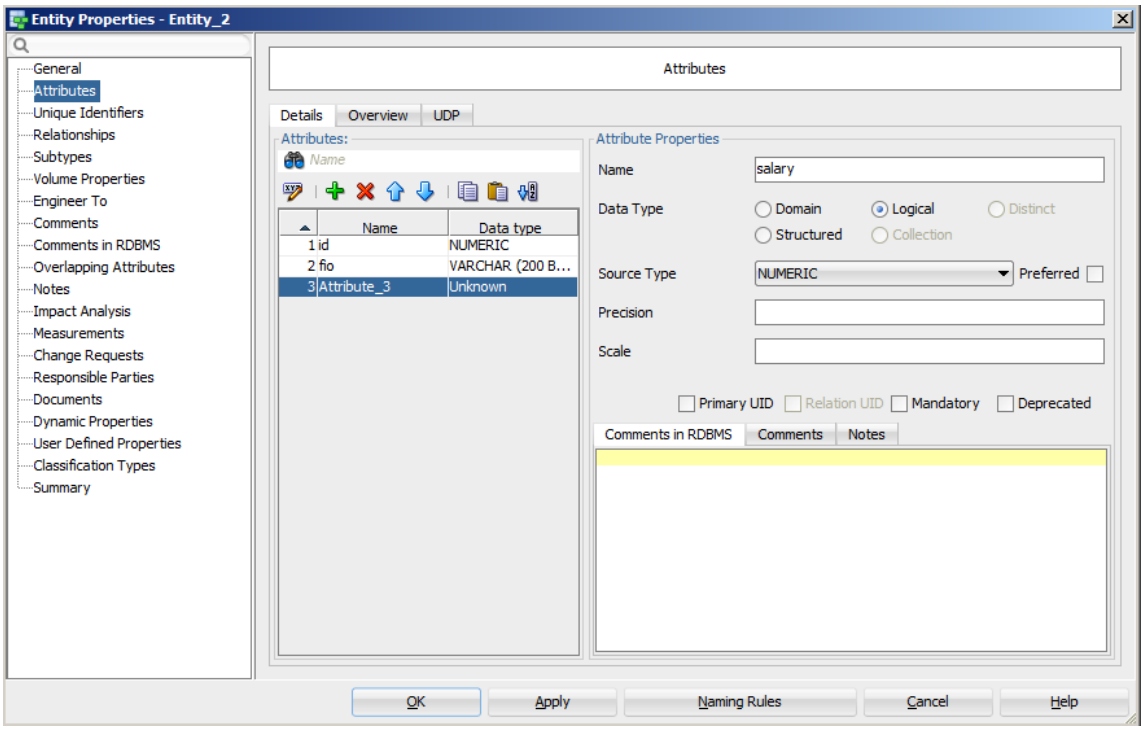
Добавляем сущность «Сотрудник»:



Атрибуты:

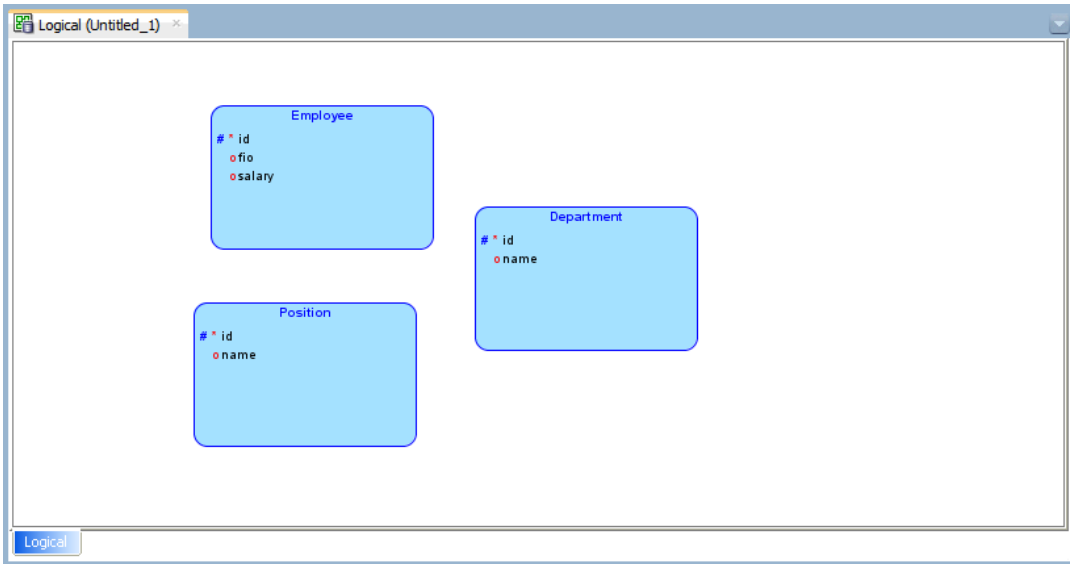






Внешние ключи не добавляем. Они будут добавлены автоматически после установки связи.

Результат:

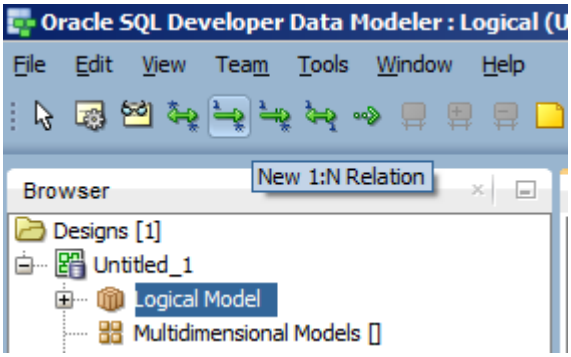


Расположение сущностей на схеме относительно друг друга выбирается произвольно.

Еще одним этапом является создание связей.

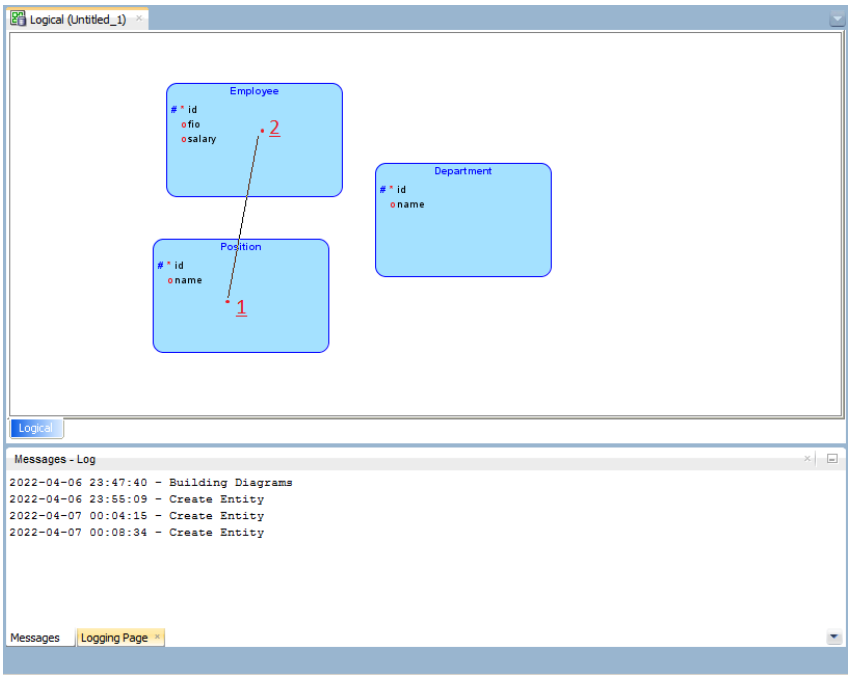
Будем полагать, что любой сотрудник имеет какую-то одну должность и работает в каком-либо одном отделе. При этом в отделе может быть любое количество сотрудников. Соответственно, речь идет о связи «один-ко-многим».

Выбираем этот тип связи на панели инструментов:

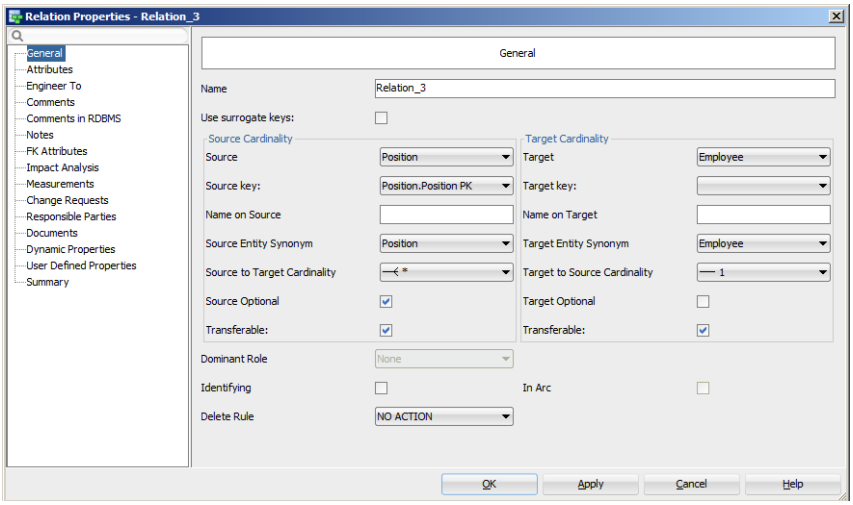


После этого нажимаем сначала на ту сущность, поле из которой будет добавлено (например, «Position»).

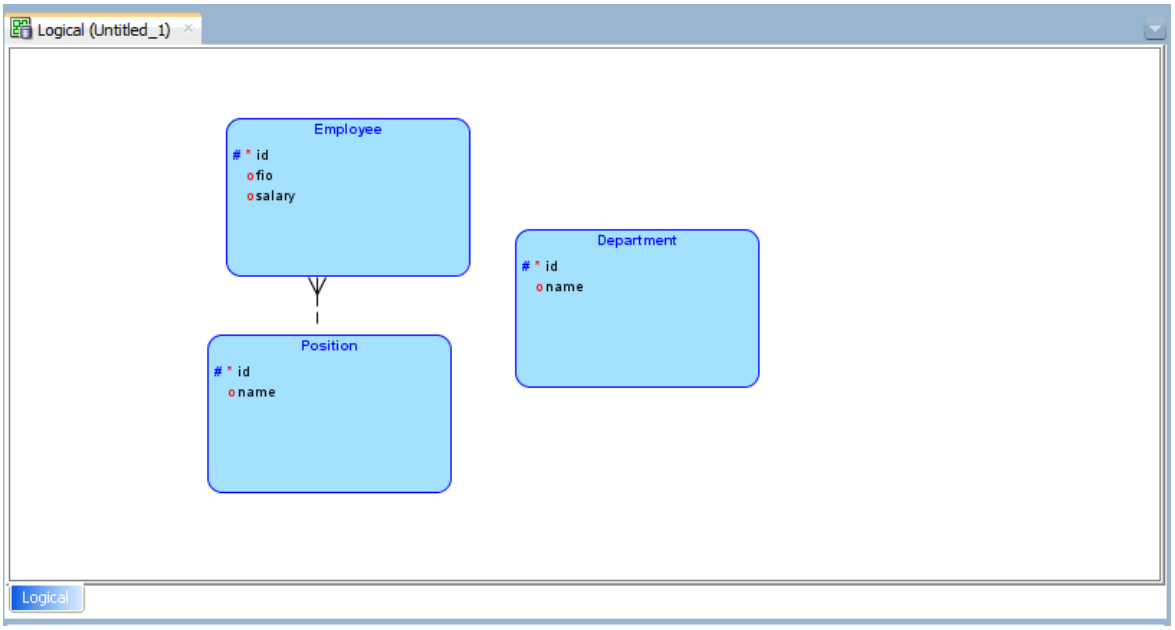
После этого нажимаем на сущность, в которую будет добавлено поле в качестве внешнего ключа («Employee»).



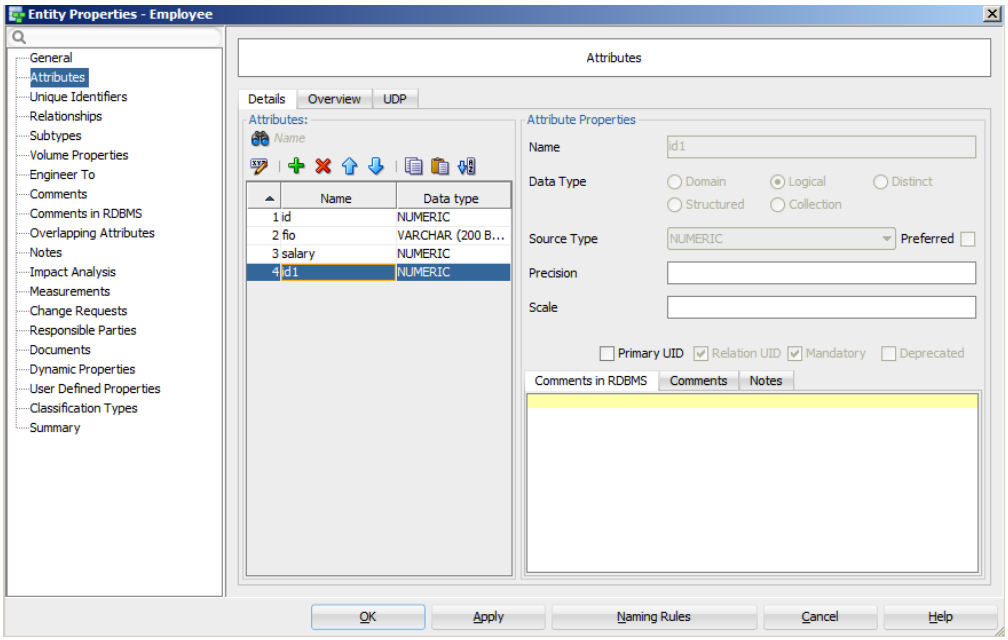
В появившемся окне можно ничего не настраивать.



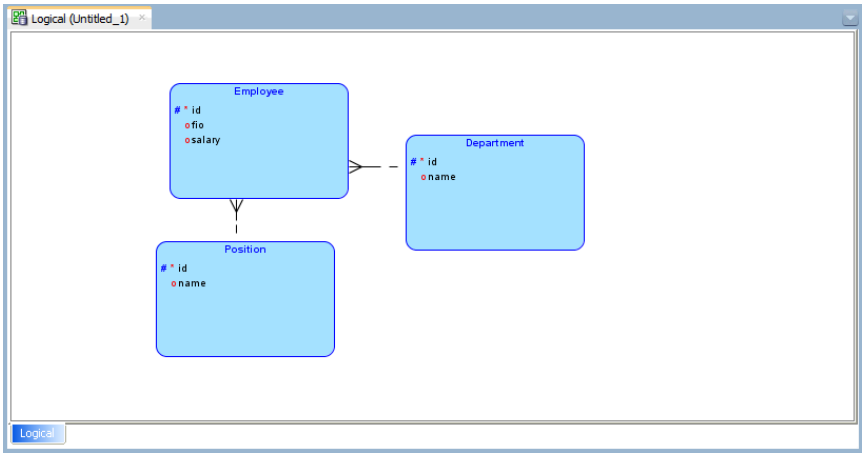
Результат:



Если открыть окно свойств сущности «Employee», то можно заметить появление нового атрибута:

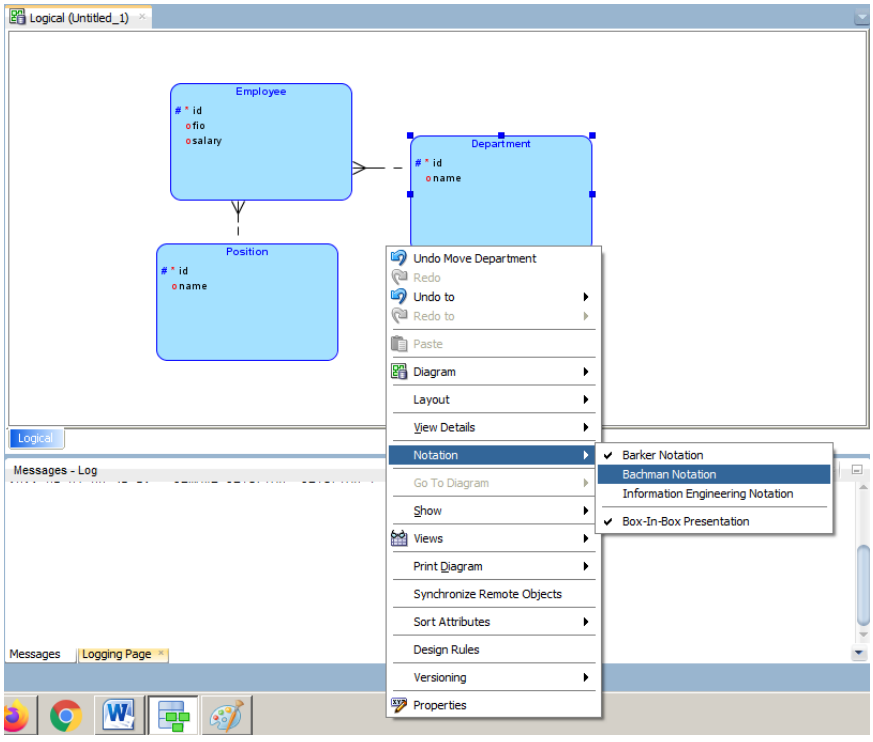


Аналогично добавляем вторую связь:

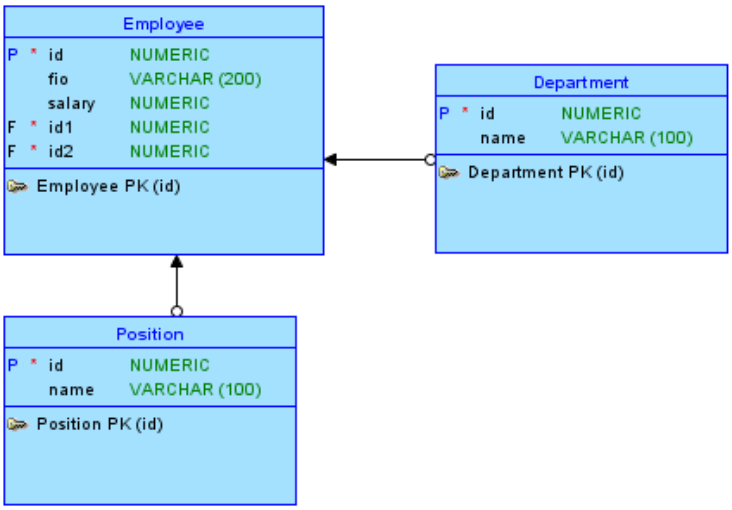


В итоге была создана логическая модель базы данных.

В некоторых случаях удобнее сменить ее внешний вид на изображение в нотацию Баркера (это можно сделать через контекстное меню).

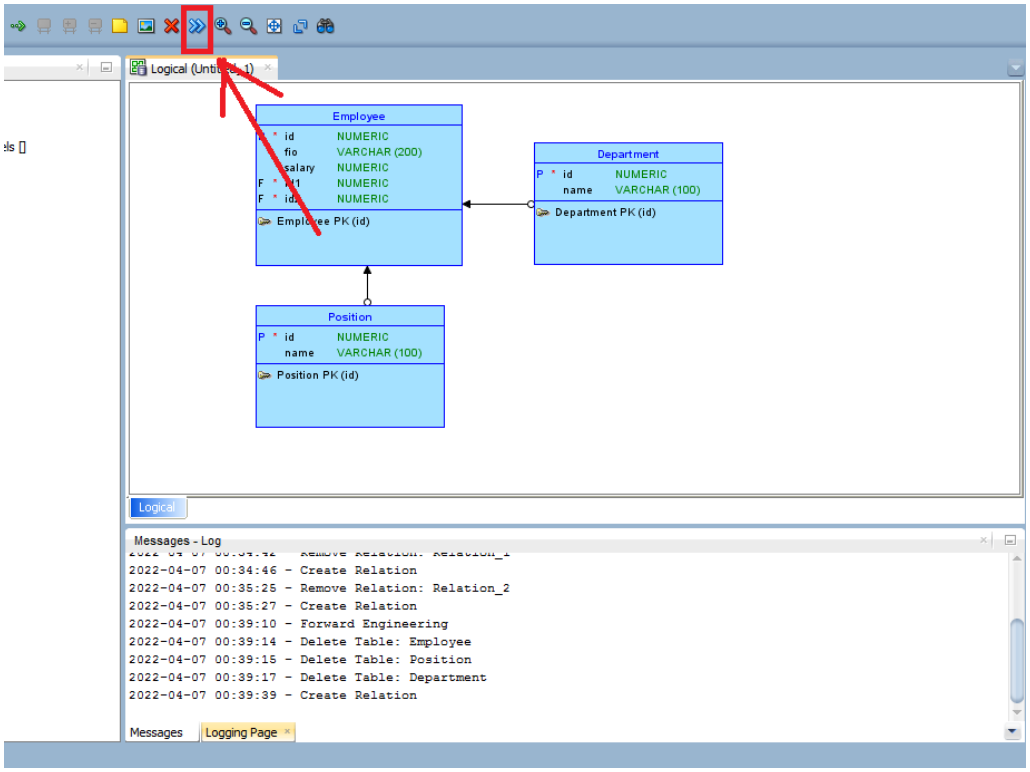


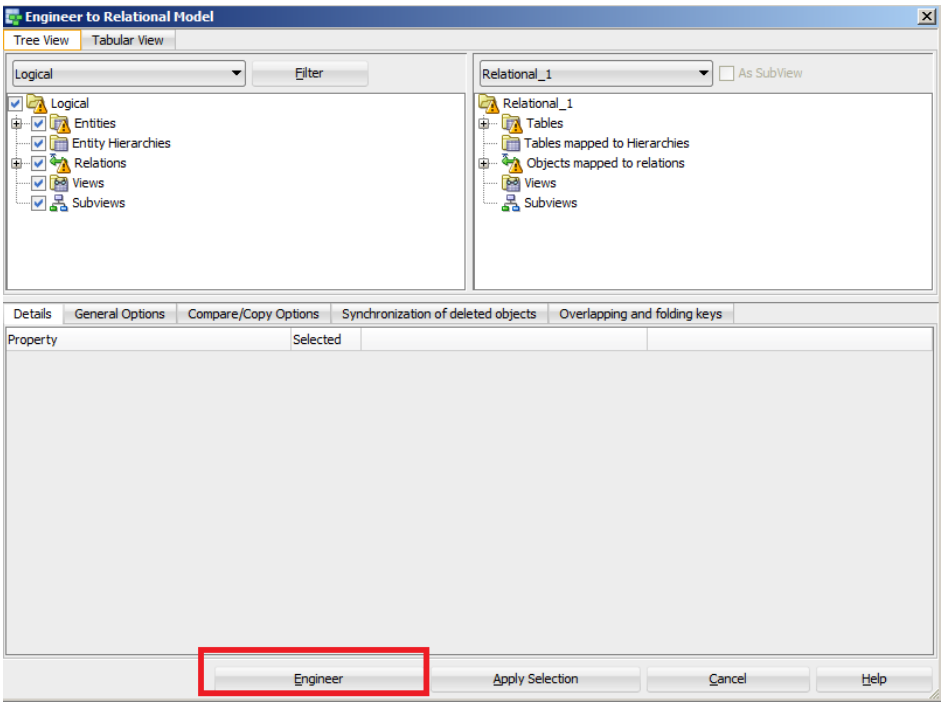
Результат:



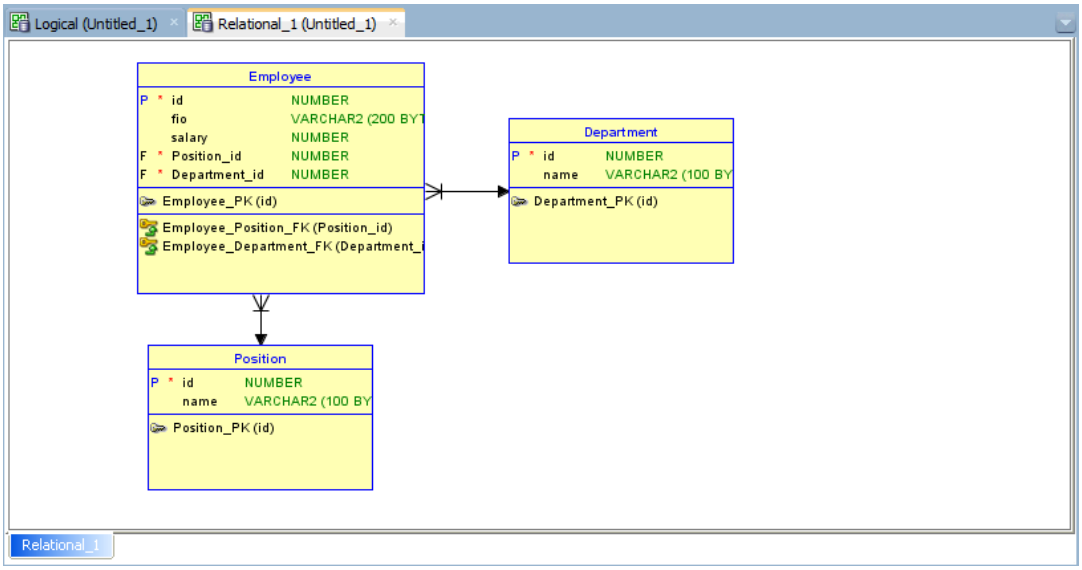
В случае дальнейшего использования данной диаграммы в технической документации поля id1 и id2 нужно будет переименовать.

Для получения реляционной модели базы данных воспользуемся встроенным инструментом:





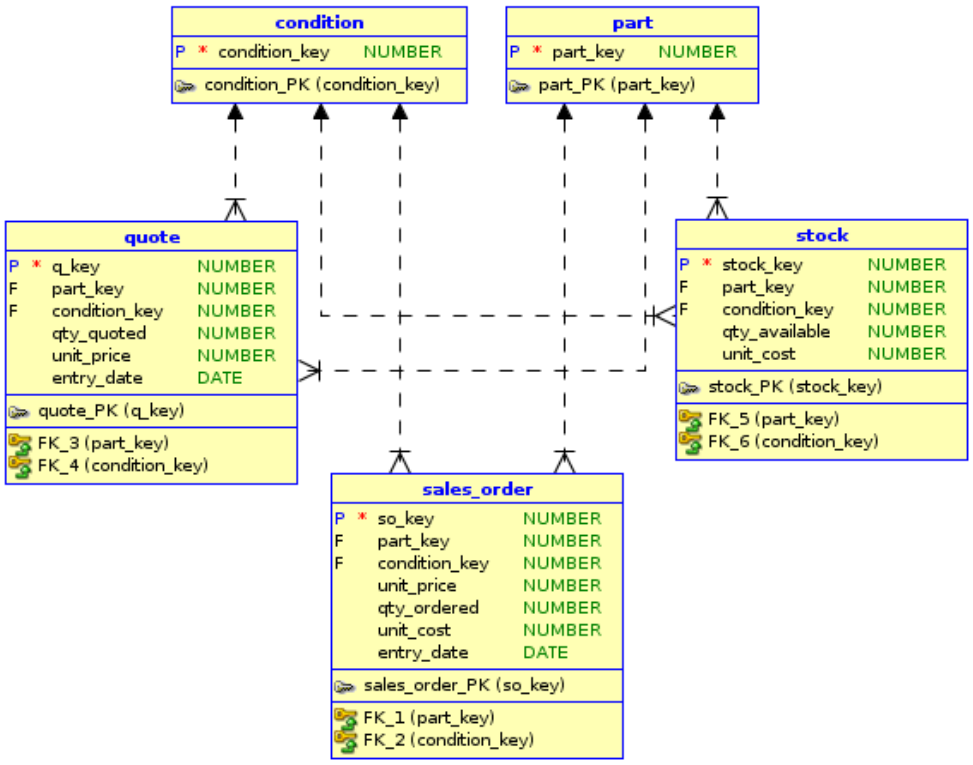
Итог:



Задание:

а) проделать все описанные выше этапы. Результат сохранить в стандартном формате Oracle SQL Developer Data Modeler и прислать в виде архива zip.

б) построить модель, представленную на рисунке. Результат сохранить в стандартном формате Oracle SQL Developer Data Modeler и прислать в виде архива zip.



Практическая работа №3 «Объекты, атрибуты и уникальные идентификаторы»

Задание 1. Определить объекты и их атрибуты для базы данных системы управления образовательным учреждением.

Принципы работы системы:

a. Школа или университет имеет множество подразделений, где учащимся преподаются курсы в течение определенного учебного периода.

b. Каждый из этих курсов ведут преподаватели.

c. Учащиеся регистрируются на различные курсы в течение учебного периода.

d. Помимо регистрационных данных, школа или университет хранит информацию о родителях учащихся.

e. Каждое подразделение ведет учет посещаемости, которая является критерием допуска к экзаменам за данный учебный период.

f. В конце каждого учебного периода проводятся экзамены, и результаты экзаменов сообщаются учащимся в установленные сроки.

g. В целях отчетности подразделение ведет журнал входов и выходов из системы, выполняемых преподавателями.

Создайте реляционную модель базы данных для системы управления образовательным учреждением.

Задание 2. Определить уникальные идентификаторы и соответствующие первичные ключи на основе предложенных сценариев.

1. Как найти определенную композицию в коллекции? Каким будет уникальный идентификатор объекта SONG?

2. У каждого учащегося в аудитории есть определенные черты или атрибуты. Какой атрибут (или атрибуты) вы выберете, чтобы выделить определенного учащегося среди всех остальных?

3. Каждого объекта из диаграммы системы управления образовательным учреждением в предыдущем задании.

В качестве ответа на данное задание необходимо представить:

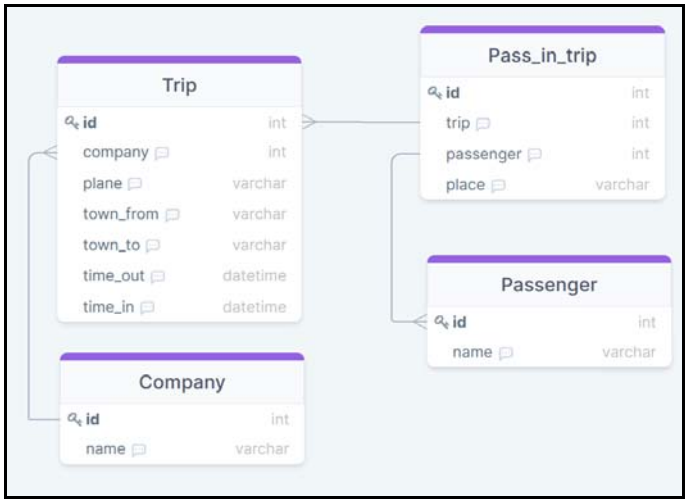
а) файл Microsoft Word с описанием объектов, из которых должна состоять спроектированная база данных (задание 1), и ответы на вопросы (задание 2).

б) архив с файлами диаграммы в Oracle SQL Developer Data Modeler (задание 1).

Практическая работа №4 «Выполнение SQL-запросов»

Ознакомиться с решением задач:

1. Вывести имена всех когда-либо обслуживаемых пассажиров авиакомпаний (https://sql-academy.org/ru/trainer/tasks/1):



Ответ:

```
SELECT name FROM Passenger
```

2. Вывести все рейсы, совершенные из Москвы (https://sql-academy.org/ru/trainer/tasks/3):

```
SELECT * FROM Trip WHERE town_from = 'Moscow'
```

3. Вывести имена людей, которые заканчиваются на "man" (https://sql-academy.org/ru/trainer/tasks/4):

```
SELECT name FROM Passenger WHERE name LIKE '%man'
```

4. Какие компании организуют перелеты с Владивостока (Vladivostok)? (https://sql-academy.org/ru/trainer/tasks/9)

```
SELECT DISTINCT Company.name FROM Company, Trip WHERE (Company.id = Trip.company) AND (town_from='Vladivostok')
```

5. Вывести количество рейсов, совершенных на TU-134 (https://sql-academy.org/ru/trainer/tasks/5)

```
SELECT COUNT(*) AS count FROM Trip WHERE plane='TU-134'
```

6. Удалить все перелеты, совершенные из Москвы (https://sql-academy.org/ru/trainer/tasks/56)

```
DELETE FROM Trip WHERE town_from = 'Moscow'
```

7. Измените имя "Andie Quincey" на новое "Andie Anthony" (https://sql-academy.org/ru/trainer/tasks/53)

```
UPDATE FamilyMembers SET member_name='Andie Anthony' WHERE member_name='Andie Quincey'
```

Самостоятельно решить задачи:

1. https://sql-academy.org/ru/trainer/tasks/2
2. https://sql-academy.org/ru/trainer/tasks/22
3. https://sql-academy.org/ru/trainer/tasks/31
4. https://sql-academy.org/ru/trainer/tasks/6
5. https://sql-academy.org/ru/trainer/tasks/34
6. https://sql-academy.org/ru/trainer/tasks/54
7. https://sql-academy.org/ru/trainer/tasks/59
8. https://sql-academy.org/ru/trainer/tasks/35
9. https://sql-academy.org/ru/trainer/tasks/38

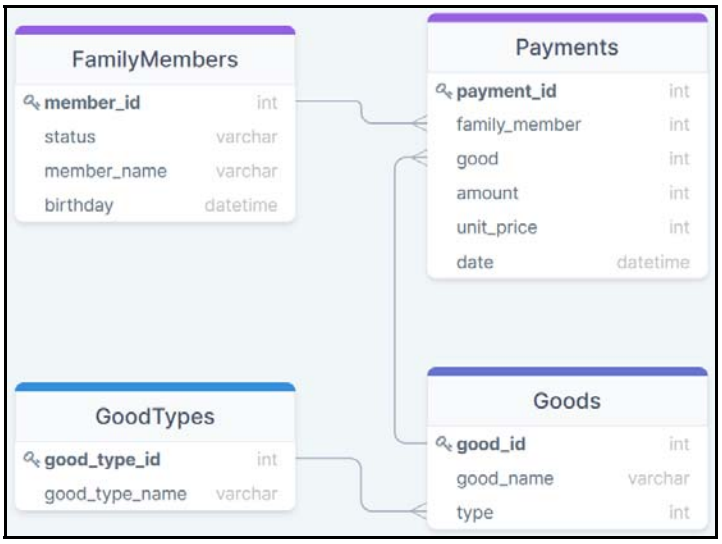
Практическая работа №5 «Выполнение SQL-запросов - 2»

Ознакомиться с решением задач:

1. Вывести имена людей, которые заканчиваются на "man" (https://sql-academy.org/ru/trainer/tasks/4):

```
SELECT name FROM Passenger WHERE name LIKE '%man'
```

2. Узнать, кто старше всех в семье (https://sql-academy.org/ru/trainer/tasks/18)




```
SELECT member_name FROM FamilyMembers WHERE birthday IN (SELECT MIN(birthday) FROM FamilyMembers)
```

Вывести имена всех членов семьи большими буквами:

```
SELECT UCASE(MEMBER_NAME) FROM FAMILYMEMBERS
```

Вывести данные о членах семьи в формате «NAME (STATUS)».
Например, «Andie Anthony (father)».

```
SELECT CONCAT(member_name, " (", status, ")") FROM FamilyMembers
```

3. Вычислить $\sqrt{169}$

```
SELECT SQRT(169)
```

4. Найти $|-20|$

```
SELECT abs(-20)
```

5. Найти остаток от деления 8 на 3

```
SELECT MOD(8, 3)
```

6. Найти натуральный логарифм числа 10

```
SELECT LOG(10)
```

7. Найти синус прямого угла

```
SELECT SIN(PI()/2)
```

8. Определить позицию буквы «Е» в строке «ПРИВЕТ ВСЕМ»

```
SELECT POSITION( 'Е' IN 'ПРИВЕТ ВСЕМ')
```

9. Из строки «MySQL» извлечь подстроку «SQL».

```
SELECT SUBSTRING('MySQL', 3, 3);
```

10. Вывести длину строки «Привет».

```
SELECT LENGTH("Hello")
```

Самостоятельно решить задачи из тренажера:

1. https://sql-academy.org/ru/trainer/tasks/36
2. https://sql-academy.org/ru/trainer/tasks/11

Подсказка: операция `LENGTH(name)`

3. https://sql-academy.org/ru/trainer/tasks/25

Подсказка: операция `YEAR(date)`

Для решения остальных задач можно использовать эмулятор на странице https://sql-academy.org/ru/sandbox

4. Найти $|38 - 20 - 168|$, используя числовые функции.
5. Найти остаток от деления 16 на 5.
6. Вычислить $\sqrt{22500}$
7. Вычислить $\log_2 16$
8. *Для базы данных «Schedule»*: вывести информацию о фамилии, имени и отчестве каждого студента в одном поле (столбце). Количество затраченных на ФИО символов – в другом.
9. *Для базы данных «Schedule»*: вывести информацию об учителях в формате «Фамилия И. О.».
10. *Для базы данных «Airbnb»*: вывести список всех почтовых сервисов, электронная почта на которых использовалась при регистрации пользователя.

Считать, что сервис – это то, что указано после @.

Например, для «cgarcia@yahoo.ca» сервисом будет «yahoo.ca».

В качестве ответа сдать отчёт в виде файла формата Microsoft Word, содержащий: текст каждой задачи и ответ в виде запроса.

Практическая работа №6 «Выполнение SQL-запросов - 3»

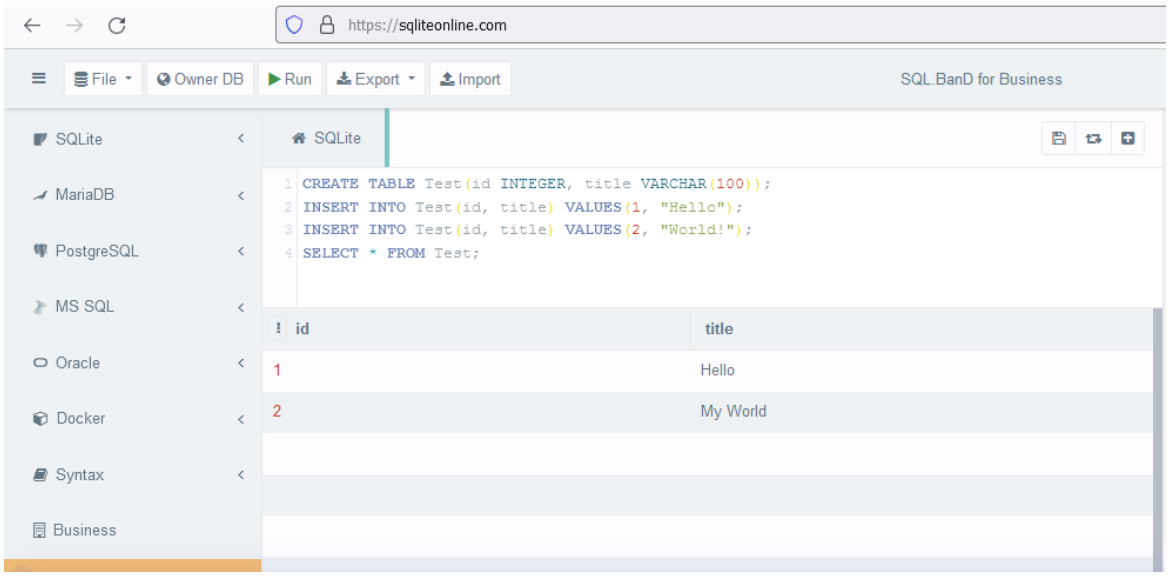
Для выполнения заданий можно использовать web-сервис https://sqliteonline.com/

1) Создать таблицу Test с полями id и title. Добавить строки "Hello" и "World!". Отобразить содержимое таблицы.

Решение:

```
CREATE TABLE Test(id INTEGER, title VARCHAR(100));
INSERT INTO Test(id, title) VALUES (1, "Hello");
INSERT INTO Test(id, title) VALUES (2, "World!");
SELECT * FROM Test;
```

Результат:



2) Создать таблицу Parent с полями id, name и tel. Заполнить данными (не менее 3-х строк). Отобразить содержимое таблицы.

Решение:

```
CREATE TABLE Parent(id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(100), phone VARCHAR(11));
INSERT INTO Parent(name, phone) VALUES ("Ivanov Pavel Ivanovich", "89188453754");
INSERT INTO Parent(name, phone) VALUES ("Bolotova Elena Nikolayevna", "89284853754");
INSERT INTO Parent(name, phone) VALUES ("Petrov Denis Maximovich", "89068457354");
SELECT * FROM Parent;
```

3) Удалить таблицу Test.

```
DROP TABLE Test;
```

4) Вывести имена родителей и номера их телефонов с указанием мобильного оператора (8918…, 8919… - МТС, 8928… - МЕГАФОН, 8905…, 8906…, 8909… – БИЛАЙН):

```
SELECT name, phone,
CASE
WHEN (phone like '8918%') or (phone like '8919%') THEN 'МТС'
WHEN phone like '8928%' THEN 'МЕГАФОН'
WHEN (phone like '8905%') or (phone like '8906%') THEN 'БИЛАЙН'
ELSE phone
END
AS MobileOperator
FROM Parent
```

5) Удалить столбец phone из таблицы Parent

```
ALTER TABLE Parent DROP phone;
SELECT * FROM Parent;
```

Задания для самостоятельного выполнения:

1) Создать таблицу marks, содержащую ФИО обучающегося и его оценку в виде целого числа от 2 до 5. Заполнить данными (как минимум 8 строк). Вывести список обучающихся с оценками и указать значение по системе «зачет-незачет» (для оценок 5 и 4 – «зачет», для остальных – «незачет»).

2) Вывести список обучающихся с оценками и их буквенное обозначение (5 – «отлично», 4 – «хорошо», 3 – «удовлетворительно», 2 – «неудовлетворительно»).

3) Удалить таблицу marks.

4) Создать таблицу преподавателей teachers, содержащую их ФИО, возраст и информацию об образовании. Заполнить данными (как минимум 8 строк). Вывести список преподавателей в возрасте до 25 лет.

5) Создать в базе данных таблицу:

№ комнаты	**Этаж**	**Категория**	**Статус**	**ЦенаРубСутки**
1	1	Номер I категории	Занят	900
2	1	Номер I категории	Занят	900
3	3	Студия	Свободен	4200
4	6	Люкс	Свободен	9000
5	8	Сюит	Свободен	19500
6	5	Люкс	Занят	9200
7	2	Номер III категории	Занят	3900
8	7	Апартамент	Занят	15100
9	2	Студия	Свободен	4500
10	1	Номер II категории	Свободен	2500
11	4	Студия	Занят	4200

Вывести все свободные номера.

6) Вывести всю имеющуюся информацию о комнатах, но вместо цены за сутки указать стоимость на неделю (произведение цены на количество дней). Вычисляемому столбцу дать имя «Цена за неделю».

7) Вывести всю имеющуюся информацию о комнатах, вычислив при этом налог на добавленную стоимость (столбец «Цена»), который включен в цену и составляет 18%, а также цену комнаты без него.

8) Для каждой комнаты установить скидку следующим образом: если этаж ниже 2, то скидка на него составит 10%. Если этаж находится между 2 и 5 (включительно), то скидка составит 20%. В остальных случаях скидка составит 25%. При выводе информации о номерах указать - у какой комнаты какая скидка.

9) Удалить столбец «Статус».

10) Опишите, каким образом стоит произвести декомпозицию последней таблицы, чтобы обеспечить более оптимальное хранение информации.

Практическая работа №7 «SQLite3»

1. В соответствии с выбранной предметной областью, создать базу данных средствами SQLite3. В качестве ответа на данное задание необходимо представить файл базы данных SQLite3, а также консольную программу на языке программирования Python, которая создает указанный файл при запуске.
2. Создать графическую программу на Python для просмотра любой из таблиц вашей базы данных в SQLite.

Предметные области:

1. Издательство.

Издательство – предприятие, занимающееся выпуском разнообразной печатной продукции. Издательство заключает договор с заказчиком (клиентом) на выполнение заказа. Заказчиком может выступать частное лицо или организация. Частное лицо может быть автором издания (или одним из авторов, если их несколько) или представителем автора. Организация для контактов с издательством также имеет своего представителя – контактную персону.

Заказ может быть книгой, брошюрой, рекламным проспектом, буклетом, бюллетенем для голосования или каким-либо другим видом издательской продукции. Подготовленные издательством материалы заказчика печатаются в типографиях, где издательство размещает свои заказы.

2. Стоматологическая поликлиника.

Поликлиника ведет прием и учет пациентов, учет их посещений (визитов) и учет обслуживания пациентов специалистами (врачами) поликлиники. Существует необходимость в хранении информации обо всех посещениях поликлиники пациентами и о том, на приеме у каких специалистов они находились.

3. Автосалон.

Существует некоторая фирма, торгующая автомобилями. Автомобиль выступает в качестве товара и как товар имеет определенные характеристики. Кроме того, на каждый автомобиль имеются технические данные. Фирма имеет своих клиентов – покупателей автомобилей, сведения о которых хранит в течение определенного времени.

4. Продажа подержанных автомобилей.

Фирма по продаже подержанных автомобилей работает с физическими лицами (клиентами фирмы), желающими купить подержанный автомобиль или автомобили. Непосредственной продажей автомобилей занимаются сотрудники фирмы – дилеры. При продаже автомобиля фирма заключает с клиентом договор, содержащий данные о клиенте, необходимые сведения об автомобиле, а также данные о дилере, обслуживающем этот договор.

5. Ассоциация фермерских хозяйств.

Предполагается, что существует некая региональная организация, назовем ее условно ассоциацией, которая является организационным объединением крестьянских фермерских хозяйств (КФХ). Ассоциация ведет учет зарегистрированных фермерских хозяйств, собирает информацию о видах их деятельности, а также о предлагаемой хозяйствами продукции и ее цене, личных данных фермеров. Вид деятельности хозяйства определяет его специализацию, например: овощеводство, животноводство, виноградарство и другие. В каждой специализации имеются виды производимых товаров.

6. Перевозки на внутригородских маршрутах.

Муниципальное автопредприятие осуществляет пассажирские перевозки на внутригородских маршрутах. Автопредприятие имеет парк автобусов, которые работают на определенных маршрутах. Работу автопредприятия обеспечивает персонал предприятия, который можно разделить по категориям занимаемых должностей на администрацию, инженерно-технический персонал и персонал, обслуживающий маршруты (водители, кондукторы). Выезжая на маршрут, водитель автобуса получает маршрутный лист (или путевой лист), содержащий данные об автобусе, маршруте, режиме работы, водителе, кондукторе.

7. Гостиница.

Предполагаемая гостиница располагает номерами с разным уровнем сервиса, комфортности и, соответственно, оплаты. Одной из характеристик номера является его тип. Предположим, что существуют следующие типы номеров: люкс – многокомнатный номер с высоким уровнем сервиса, комфортности и обслуживания; полулюкс – номер меньшей, чем люкс, площади, но с достаточным уровнем сервиса и комфортности; одноместный, двухместный номер с минимальным уровнем сервиса; многоместный номер, также с минимальным уровнем сервиса. Стоимость для номеров типа люкс и полулюкс устанавливается как стоимость всего номера (в сутки), независимо от количества проживающих в номере. Стоимость проживания в одно-, двух- и много-местных номерах устанавливается для одного человека (в сутки). Номера и места в номерах могут бронироваться. При наличии телефона в номере пользование междугородным телефоном оплачивается отдельно по фактическим счетам.

Все прибывающие и размещаемые в гостинице граждане при вселении должны заполнить карточку регистрации. Кроме того, для расчетов с клиентами администрация гостиницы заводит расчетные карточки, которые содержат данные о размещении клиента, оказанных услугах и всех расчетах с ним. Для номеров типа люкс и полулюкс расчетная карточка может заводиться только для одного клиента, оплачивающего номер.

Любой номер гостиницы имеет номер, по которому ведется учет проживающих в гостинице. Это свойство номера в последующих описаниях будем называть номером комнаты (независимо от фактического количества комнат в номере). При выбытии клиента данные о нем сохраняются в архиве.

8. Бюро знакомств.

База потенциальных женихов и невест: пол, регистрационный номер, дата регистрации, сведения о себе (возраст, рост, вес, знак зодиака и т.д.), требования к партнеру (возраст, рост, вес, знак зодиака и т.д.). Выбор подмножества подходящих кандидатур, подготовка встреч (формирование приглашения для знакомства).

9. Биржа труда.

Информация о безработных: анкетные данные, профессия, образование, место и должность последней работы, причина увольнения, семейное положение, жилищные условия, контактные координаты, требования к будущей работе.

Информация о вакансиях: фирма, должность, условия труда и оплаты, требования к специалисту.

10. Справочник меломана.

Информация о группах и исполнителях, песен и архивах с перечнем песен (в виде ссылок).

11. Управление кухней ресторана.

Рецептуры блюд: раскладка, рецепт приготовления.

Продукты на складе: наименование, цена, количество.

Заказ меню на день (на заданное число персон); званый ужин.

12. Справочник абитуриента.

Информация о вузах: наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении.

13. Справочник селекционера.

Наименование сорта какой-либо культуры, автор, родительские сорта, урожайность, характеристики плодов, морозоустойчивость, устойчивость к вредителям и болезням, наличие в том или ином селекционном фонде.

14. Телепрограмма.

Программа телепередач нескольких телеканалов (на неделю по дням, часам). Разные жанры телепередач: новости, спорт, художественные фильмы, сериалы и т. д.

15. Справочник географа.

Города (географические координаты, численность населения), регионы (области, провинции, штаты и т. д. — принадлежность стране, столица, численность населения), страны (площадь, численность населения, форма государственного правления, столица), материки.

Практическая работа №8 «LibreOffice Base»

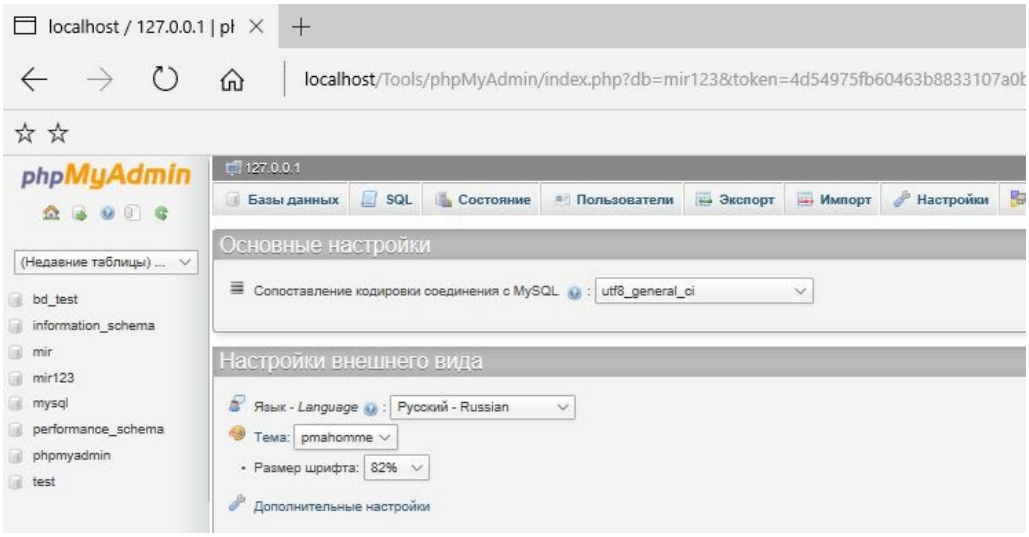
Задание 1: В соответствии с предметной областью, выбранной в предыдущей работе, определить набор таблиц, которые должны присутствовать в базе данных.

В качестве ответа на данное задание необходимо представить файл Microsoft Word с описанием таблиц (название, поля и их типы), из которых должна состоять спроектированная база данных.

Задание 2: Создать файл LibreOffice Base, содержащий описанные таблицы и формы для их заполнения. В каждой таблице добавить не менее двух записей.

Практическая работа №9 «Создание базы данных в phpMyAdmin»

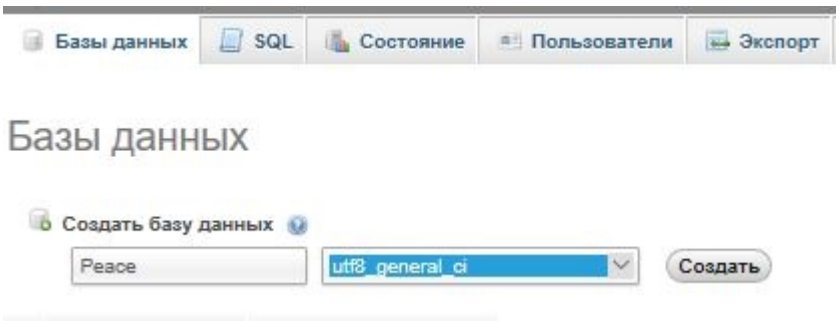
Входим на страницу администрирования базы данных MySQL Для того, чтобы зайти на страницу администрирования баз данных, в командной строке Вашего браузера введите следующий адрес: **localhost/tools/phpmyadmin/**. Вы должны увидеть примерно следующую картину:



В левой колонке находятся имеющиеся базы данных (у Вас их будет скорее всего меньше чем на скриншоте), в центральной части основные настройки (здесь Вы можете изменить язык, вид, кодировку). Верхние вкладки предназначены для различных задач (о них мы еще поговорим в процессе).

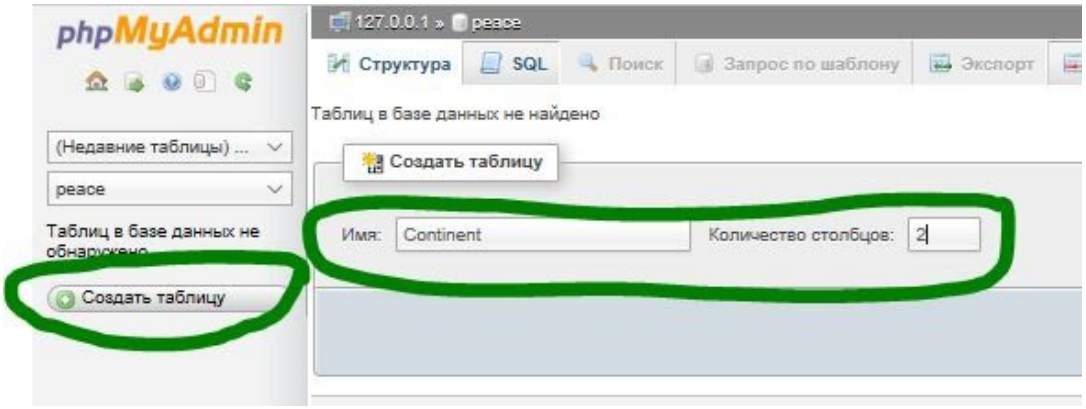
Создание базы данных MySQL

Для того, чтобы создать новую базу данных нажмем на верхнюю вкладку «Базы данных» и перед нами на центральном поле откроется список всех имеющихся баз данных MySQL. Нам же нужно создать новую. Для этого в поле «Создать базу данных» впишем название создаваемой базы и нажмем на кнопку «Создать».

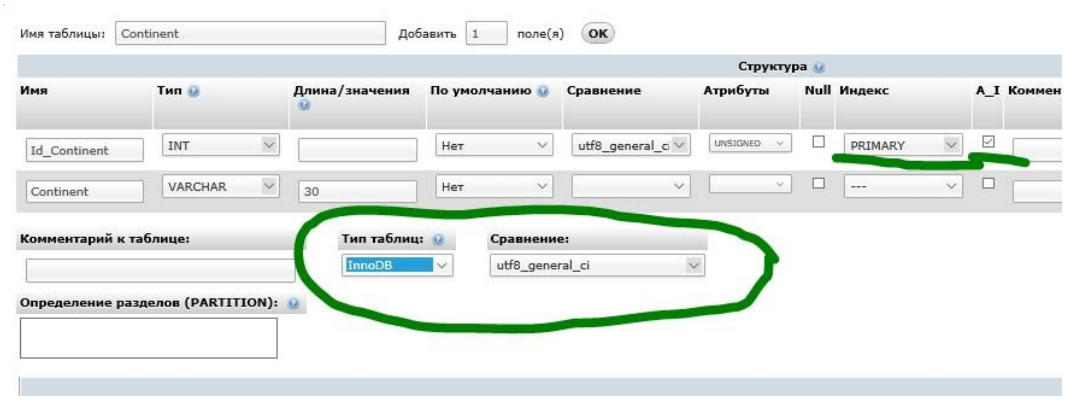


После того, как Вы нажмете на кнопку «Создать» база данных добавиться в список баз данных в панели слева и на центральном поле. Теперь выберете новую базу данных, кликнув по ее названию.

Здесь Вам будет предложено создать таблицу. **Давайте создадим таблицу базы данных.** Для этого заполните поля «Имя» и «Количество столбцов» и нажмите Ок.



После этого перед Вами откроется страница для заполнения полей новой таблицы базы данных. Здесь каждому полю нужно присвоить имя, тип хранимых данных, длину (если требуется для данного атрибута) и для такого поля как идентификатор (id) также требуется указать авто-инкремент и первичный ключ. Это должно выглядеть как на скриншоте ниже.



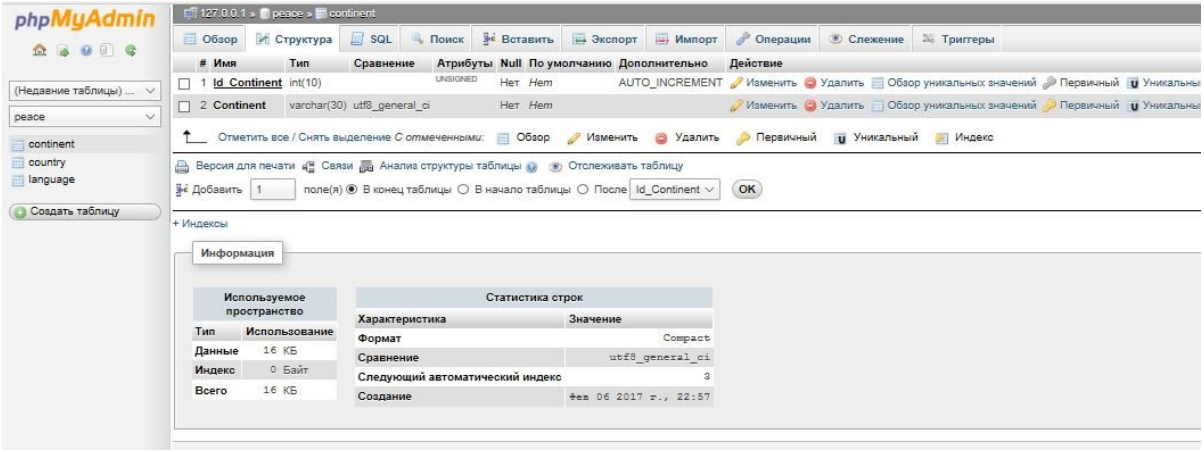
Немного о типах данных.

В моем примере нам встретились такие типы данных для таблицы базы как int и varchar.
Int – обозначает, что в данном поле будут храниться целые числа (поле id). Id будет являться идентификатором и для каждого нового элемента он будет разным. При добавлении нового элемента в таблицу, id будет увеличиваться на единицу по сравнению с предыдущим добавленным элементом (именно поэтому поле id должно иметь авто-инкремент (авто приращение) и первичный ключ). С помощью этого поля можно идентифицировать объекты, оно очень поможет нам, когда мы начнем взаимодействовать с базой данных с помощью кода php.

Varchar - тип данных, который может хранить отдельный символ или строку (длиной до 8000 символов). Для таких полей как имя, фамилия, адрес электронной почты он вполне даже подойдет. Еще ему можно задавать длину. В моем примере я задала длину 100, так как все поля в моем примере с этим типом данных не займут более 100 символов.

Существуют и разные другие типы данных, предназначенные для хранения даты, текста и других данных. Вы можете поискать информацию о типах данных в интернете. Также о них я буду рассказывать в следующих уроках.

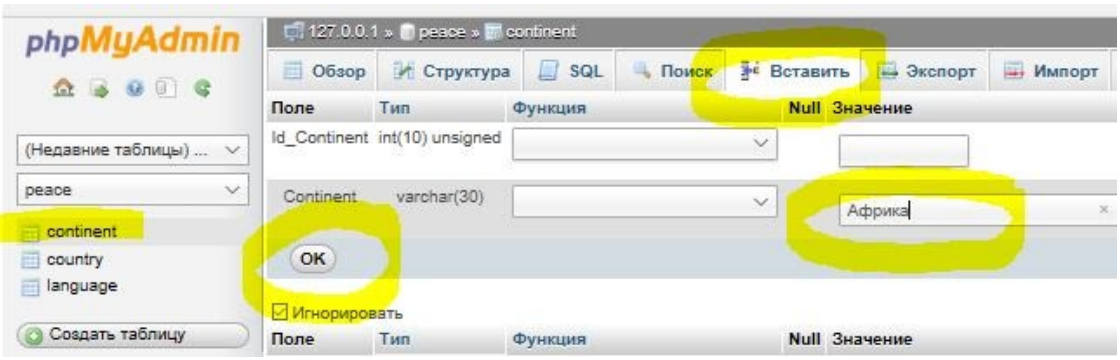
Далее нажимаем на кнопку «Сохранить» и перед нами созданная таблица, в которой пока нет ни одной записи. Таблица появиться у Вас в панели слева и на центральной части экрана. Щелкните по ее имени, чтобы увидеть структуру.



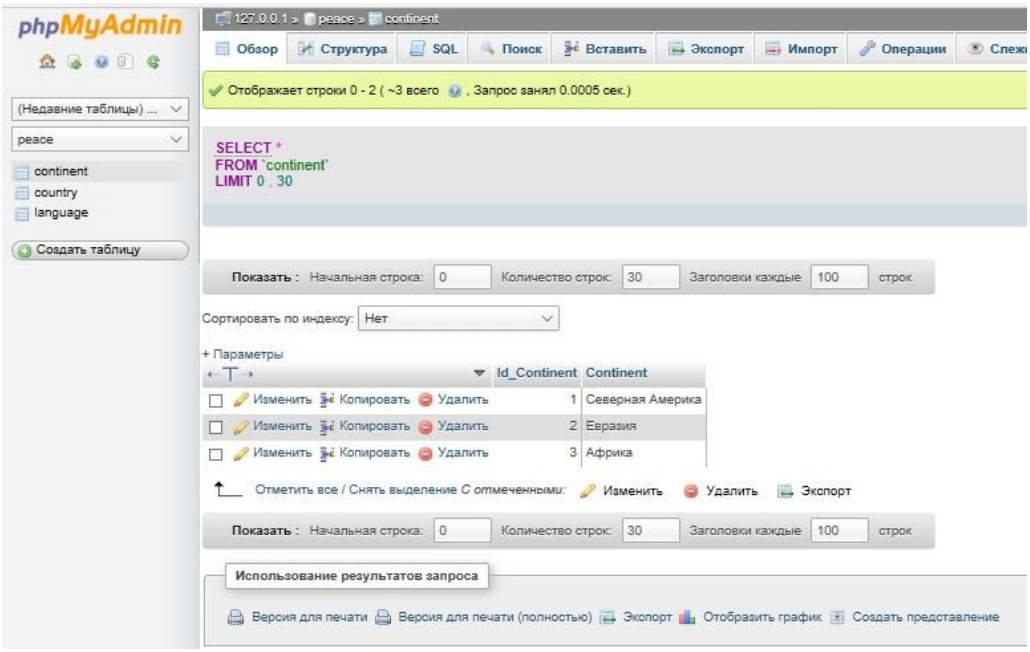
Здесь можно удалить, изменить и добавить поле. Интерфейс интуитивно понятный и сделать это не составит какого-либо труда.

Теперь посмотрим как **вставить новый элемент в таблицу базы данных**.

Для этого кликните по верхней вкладке «Вставить» и перед Вами откроется страница для вставки нового элемента в таблицу базы. Заполните все поля (кроме поля id, оно будет заполняться автоматически) и нажмите на кнопку ОК.



После того как Вы нажмете Ок, перейдите на вкладку «Обзор» (она находится вверху) и Вы увидите новый добавленный Вами элемент в Вашу таблицу базы данных MySQL.

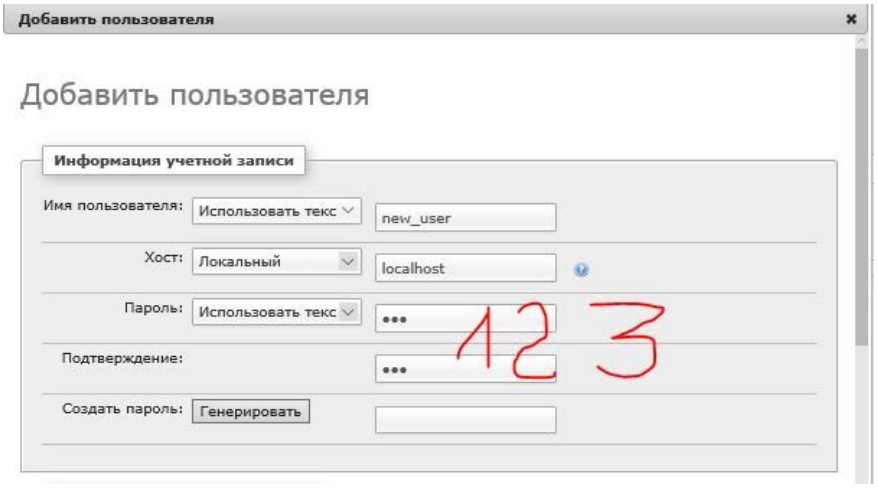


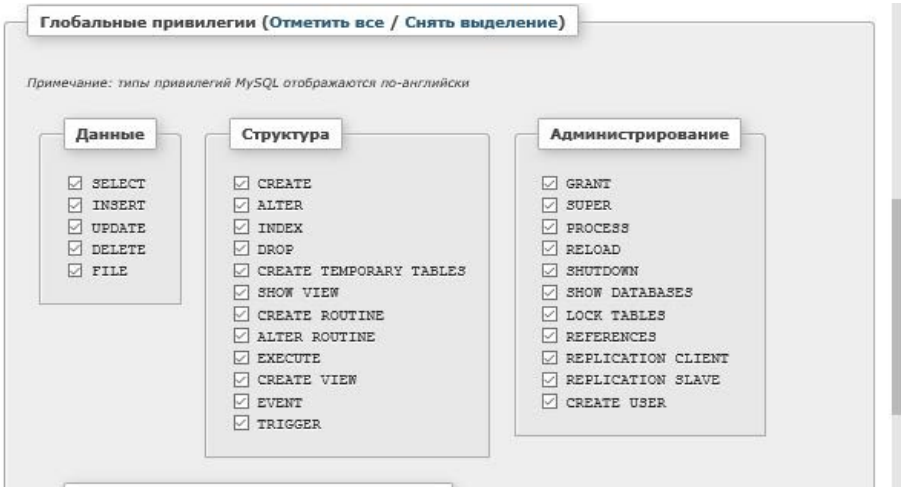
На этой же страничке Вы можете удалить или изменить добавленный элемент из таблицы базы данных.

Теперь посмотрим как создать нового пользователя для базы данных, создать ему имя и пароль и выставить ему определенные привилегии. Информация об имени пользователя и пароле понадобится нам, когда мы будем соединяться с базой при помощи php скрипта.

Итак, **создаем нового пользователя для базы данных.** Для этого нажимаем вверху на название нашей базы данных, после этого в верхних вкладках появится пункт «Привилегии», нажимаем на него.

Нажимаем «Добавить пользователя». Перед Вами откроется страница с полями, которые нужно заполнить (имя пользователя, хост, пароль и подтверждение пароля). В качестве хоста нужно выбрать локальный хост.





После чего нажимаем добавить пользователя и новый пользователь будет добавлен. Вы увидите сообщение, что добавлен пользователь к базе данных admin со всеми привилегиями.

Здесь Вы также можете редактировать привилегии, нажав на редактирование привилегий. Это может понадобиться в том случае, если кто-то еще должен иметь доступ к Вашей базе, но Вы хотите ограничить этого человека в привилегиях (например он не может удалять данные). Тогда Вы создаете нового пользователя для Вашей базы данных, но выставляете ему определенные привилегии.

Наконец, чтобы **удалить базу данных** нужно снова перейти на вкладку «Базы данных», выбрать базу для удаления и нажать на ссылочку удалить. Здесь мы рассмотрели основные операции взаимодействия с базой данных, через панели администрирования.

Создание связей между таблицами с помощью phpmyadmin

Далее научимся создавать **связи** между таблицами в базе данных **MySQL** с помощью **phpmyadmin**. Если по какой-то причине вы не желаете использовать **phpmyadmin**, смотрите приведенные ниже SQL-запросы.

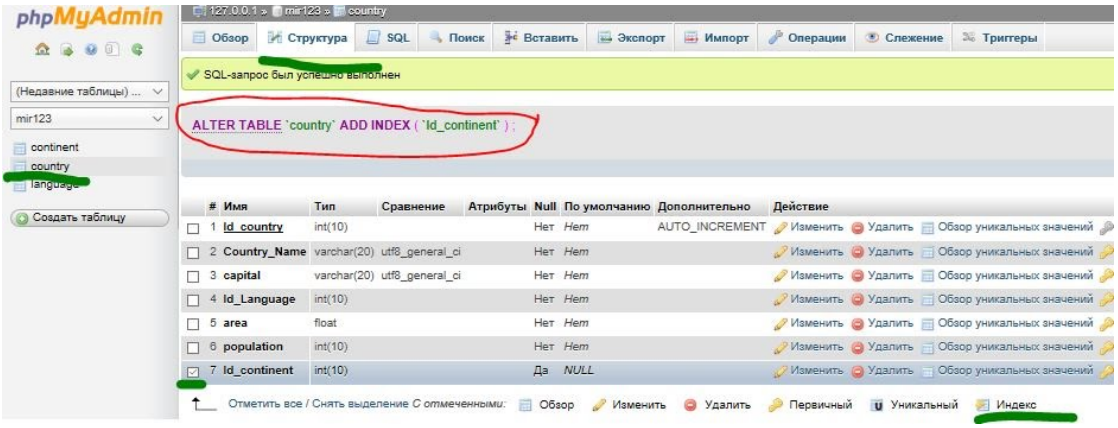
Почему же связи удобно держать в самой базе данных? Ведь эту задачу обычно решает так и само приложение? Все дело в ограничениях и действиях при изменении, которые можно наложить на связи.

Например, можно запретить удалять категорию, если с ней связана хотя б одна заметка. Или удалить все заметки, если удалена категория. Или установить NULL в связующее поле. В любом случае, с помощью связей повышается отказоустойчивость и надежность приложения.

Для начала, движок таблиц должен быть **InnoDB**. Только он поддерживает внешние ключи (**foreign key**). Если у вас таблицы **MyISAM**, почитайте как их конвертировать в InnoDB.

Для того, чтобы связать таблицы по полям, необходимо сначала **добавить в индекс** связываемые поля:

В **phpmyadmin** выбираем таблицу Country, выбираем режим структуры, выделяем поле Id_Continent, для которого будем делать внешнюю связь и кликаем Индекс.



Обратите внимание на разницу между "Индекс" и "Уникальный". Уникальный индекс можно использовать, например, до поля id, то есть там, где значения не повторяются.

Это же действие можно сделать с помощью **SQL**-запроса:

```
ALTER TABLE `table_name` ADD INDEX ( `field_name` ) ;
```

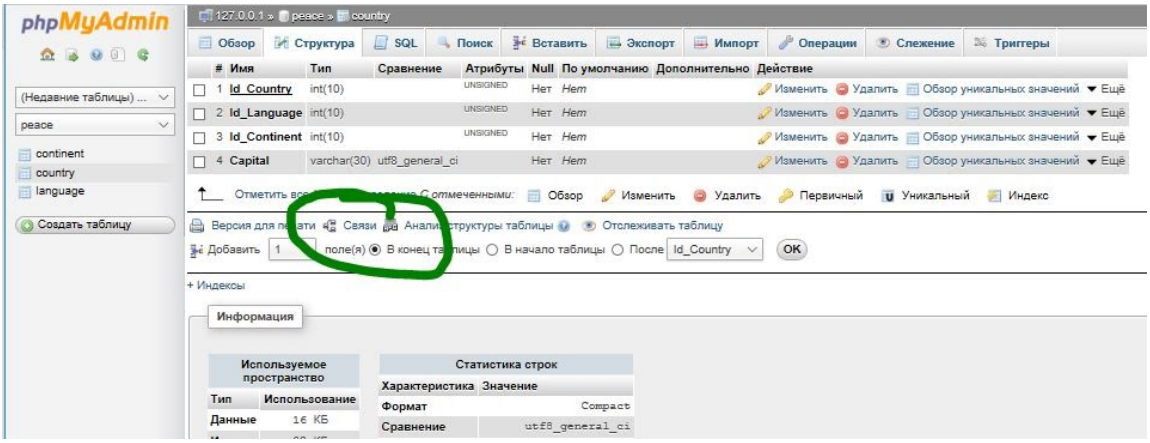
Аналогично добавляем индекс для поля Id_Language таблицы Country.

Аналогично можно добавить индекс (только в моем случае теперь уже уникальный или первичный) для таблицы, на которую ссылаемся, для поля id. Поскольку поле id у меня идентификатор, для него делаем первичный ключ. Уникальный ключ мог бы понадобится для других уникальных полей. Но так как мы его уже создали, принимаем информацию к сведению и идём дальше

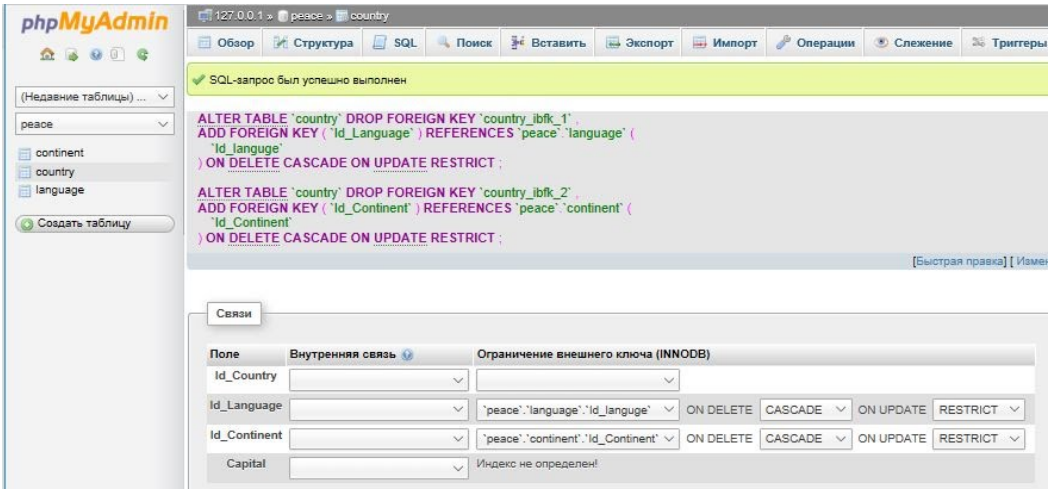
С помощью **SQL**-запроса:

```
ALTER TABLE `table_name` ADD UNIQUE (`field_name`);
```

Теперь осталось только **связать таблицы**. Для этого кликаем внизу на пункт Связи:



Теперь для доступных полей (а доступны только проиндексированные поля) выбираем связь с внешними таблицами и действия при изменении записей в таблицах:

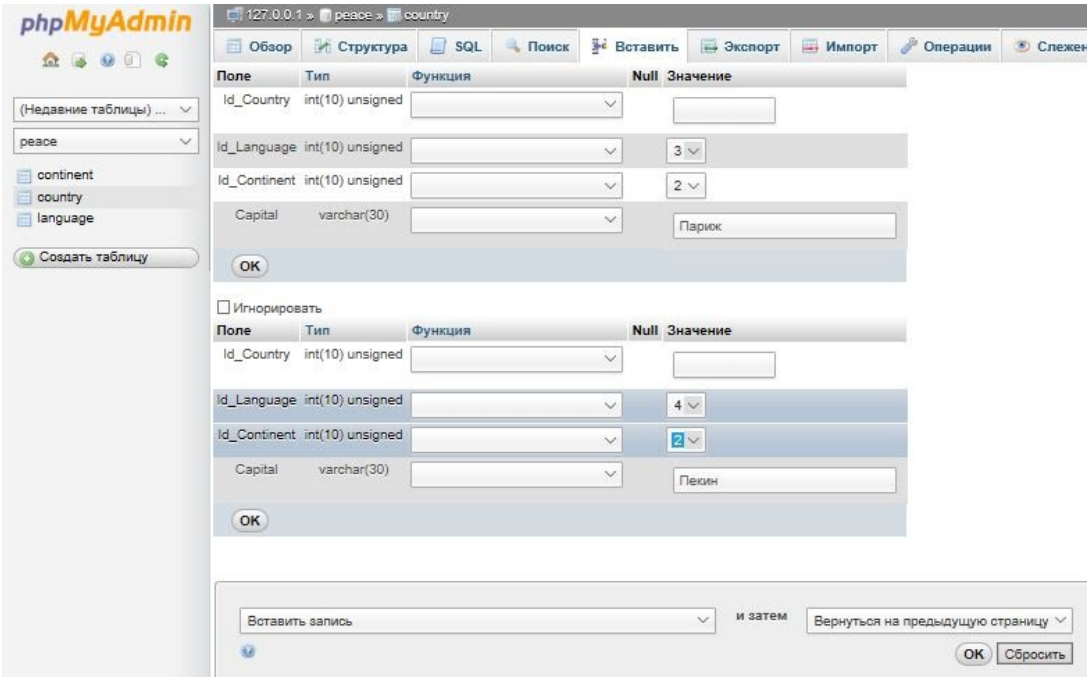


Через **SQL**-запрос:

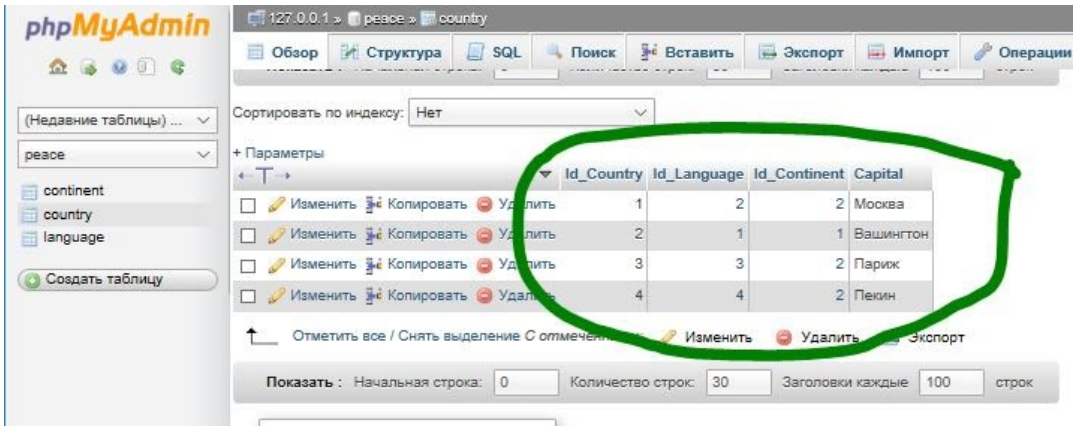
```
ALTER TABLE `table_name` ADD FOREIGN KEY ( `field_in_table_name_which_need_connect` )
REFERENCES `outer_table_to_connect` (`outer_field`) ON DELETE RESTRICT ON UPDATE
RESTRICT ;
```

на этом все, таблицы связаны через **foreign key**.

Теперь пробуем добавить записи в таблицу Country:



Результат:



Задание: Создать таблицы и заполнить их данными в соответствии с предметной областью, выбранной в практическом задании №7.

Практическая работа №10 «Python и MySQL: практическое введение»

Установка MySQL Connector/Python:

Драйвер базы данных — программное обеспечение, позволяющее приложению подключаться и взаимодействовать с СУБД. Такие драйверы обычно поставляются в виде отдельных модулей. Сандартный интерфейс, которому должны соответствовать все драйверы баз данных Python, описан в PEP 249. Драйверы баз данных Python, такие как sqlite3 для SQLite, psycopg для PostgreSQL и MySQL Connector/Python для MySQL, следуют этим правилам.

Для установки драйвера (коннектора) воспользуемся менеджером пакетов pip:

```
pip install mysql-connector-python
```

pip установит коннектор в текущую активную среду. Чтобы работать с проектом изолированным образом, мы рекомендуем настроить виртуальную среду.

Проверим результат установки, запустив в терминале Python следующую команду:

```
import mysql.connector
```

Если инструкция импорта выполняется без ошибок, значит mysql.connector успешно установлен и готов к использованию.

Установление соединения с сервером MySQL

MySQL — это серверная система управления базами данных. Один сервер может содержать несколько баз данных. Чтобы взаимодействовать с базой данных, мы должны установить соединение с сервером. Рабочий процесс программы Python, которая взаимодействует с базой данных на основе MySQL, в общих чертах выглядит следующим образом:

1. Подключаемся к серверу MySQL.
2. Создаем новую базу данных (при необходимости).

3. Соединяемся с базой данных.
4. Выполняем SQL-запрос, собираем результаты.
5. Сообщаем базе данных, если в таблицу внесены изменения.
6. Закрываем соединение с сервером MySQL.

Каким бы ни было приложение, первый шаг — связать между собой приложение и базу данных.

Подключаемся к серверу MySQL из Python

Чтобы установить соединение, используем connect() из модуля mysql.connector. Эта функция принимает параметры host, user и password, а возвращает объект MySQLConnection. Учетные данные можно получить в результате ввода от пользователя:

```
from getpass import getpass
from mysql.connector import connect, Error

try:
    with connect(
        host="localhost",
        user=input("Имя пользователя: "),
        password=getpass("Пароль: "),
    ) as connection:
        print(connection)
except Error as e:
     print(e)
```

Объект MySQLConnection хранится в переменной connection, которую мы будем использовать для доступа к серверу MySQL. Несколько важных моментов:

- Все соединения с базой данных оборачивайтев блоки try ... except. Так будет проще перехватить и изучить любые исключения.
- Не забывайте закрывать соединение после завершения доступа к базе данных. Неиспользуемые открытые соединения приводят к неожиданным ошибкам и проблемам с производительностью. В коде для этого используется диспетчер контекста (with ... as ...).
- Никогда не следует встраивать учетные данные (имя пользователя и пароль) в строковом виде в скрипт Python. Это плохая практика для развертывания, которая представляет серьезную угрозу безопасности. Приведенный код запрашивает для входа учетные данные. Для этого используется встроенный модуль getpass, чтобы скрыть вводимый пароль. Хотя это лучше, чем жесткое кодирование, но есть и другие, более безопасные способы хранения конфиденциальной информации, например, использование переменных среды.

Итак, мы установили соединение между нашей программой и сервером MySQL. Теперь нужно либо создать новую базу данных, либо подключиться к существующей.

Создаем новую базу данных

Чтобы создать новую базу данных, например, с именем online_movie_rating, нужно выполнить инструкцию SQL:

```
CREATE DATABASE online_movie_rating;
```

MySQL обязывает ставить точку с запятой (;) в конце оператора. Однако MySQL Connector/Python автоматически добавляет точку с запятой в конце каждого запроса.

```
try:
    with connect(
        host="localhost",
        user=input("Имя пользователя: "),
        password=getpass("Пароль: "),
    ) as connection:
        create_db_query = "CREATE DATABASE online_movie_rating"
        with connection.cursor() as cursor:
            cursor.execute(create_db_query)
except Error as e:
     print(e)
```

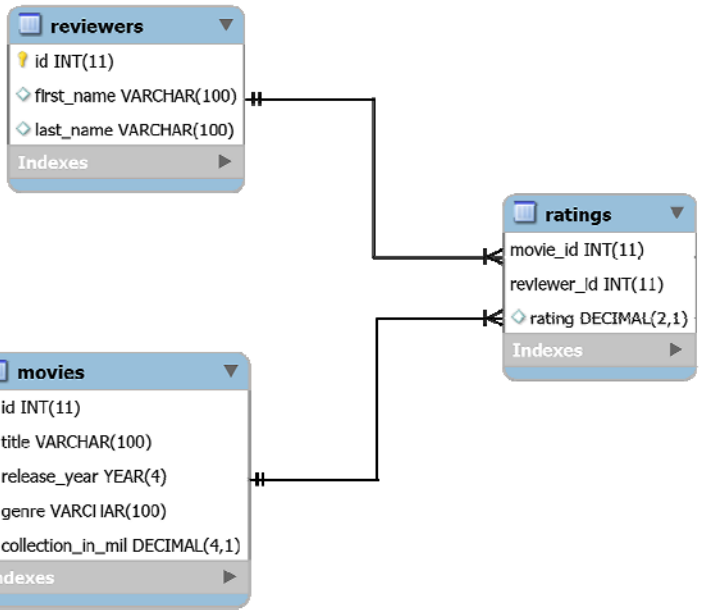
Запрос CREATE DATABASE сохраняется в виде строки в переменной create_db_query, а затем передается на выполнение в cursor.execute().

Если база данных с таким именем уже существует на сервере, мы получим сообщение об ошибке. Используя тот же объект MySQLConnection, что и ранее, выполним запрос SHOW DATABASES, чтобы увидеть все таблицы, хранящиеся в базе данных:

```
try:
    with connect(
        host="localhost",
        user=input("Введите имя пользователя: "),
        password=getpass("Введите пароль: "),
    ) as connection:
        show_db_query = "SHOW DATABASES"
        with connection.cursor() as cursor:
            cursor.execute(show_db_query)
            for db in cursor:
                print(db)
except Error as e:
     print(e)
```

Приведенный код выведет имена всех баз данных, находящихся на нашем сервере MySQL. Команда SHOW DATABASES в нашем примере также вывела базы данных, которые автоматически создаются сервером MySQL и предоставляют доступ к метаданным баз данных и настройкам сервера.

Задание: создать следующую структуру:



Пример кода для создания таблицы:

```
create_movies_table_query = """
CREATE TABLE movies(
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100),
    release_year YEAR(4),
    genre VARCHAR(100),
    collection_in_mil INT
)
"""

with connection.cursor() as cursor:
    cursor.execute(create_movies_table_query)
     connection.commit()
```

Пример кода для заполнения таблицы данными:

```
insert_movies_query = """
INSERT INTO movies (title, release_year, genre, collection_in_mil)
VALUES
    ("Forrest Gump", 1994, "Drama", 330.2),
    ("3 Idiots", 2009, "Drama", 2.4),
    ("Eternal Sunshine of the Spotless Mind", 2004, "Drama", 34.5),
    ("Good Will Hunting", 1997, "Drama", 138.1),
    ("Skyfall", 2012, "Action", 304.6),
    ("Gladiator", 2000, "Action", 188.7),
    ("Black", 2005, "Drama", 3.0),
    ("Titanic", 1997, "Romance", 659.2),
    ("The Shawshank Redemption", 1994, "Drama",28.4),
    ("Udaan", 2010, "Drama", 1.5),
```

```
        ("Home Alone", 1990, "Comedy", 286.9),
        ("Casablanca", 1942, "Romance", 1.0),
        ("Avengers: Endgame", 2019, "Action", 858.8),
        ("Night of the Living Dead", 1968, "Horror", 2.5),
        ("The Godfather", 1972, "Crime", 135.6),
        ("Haider", 2014, "Action", 4.2),
        ("Inception", 2010, "Adventure", 293.7),
        ("Evil", 2003, "Horror", 1.3),
        ("Toy Story 4", 2019, "Animation", 434.9),
        ("Air Force One", 1997, "Drama", 138.1),
        ("The Dark Knight", 2008, "Action",535.4),
        ("Bhaag Milkha Bhaag", 2013, "Sport", 4.1),
        ("The Lion King", 1994, "Animation", 423.6),
        ("Pulp Fiction", 1994, "Crime", 108.8),
        ("Kai Po Che", 2013, "Sport", 6.0),
        ("Beasts of No Nation", 2015, "War", 1.4),
        ("Andadhun", 2018, "Thriller", 2.9),
        ("The Silence of the Lambs", 1991, "Crime", 68.2),
        ("Deadpool", 2016, "Action", 363.6),
        ("Drishyam", 2015, "Mystery", 3.0)
    """
    with connection.cursor() as cursor:
        cursor.execute(insert_movies_query)
        connection.commit()
```

Код программы представить в виде файла формата «py».

Практическая работа №11 «Реализация базы данных в MySQL»

Пример задания и его выполнения:

Создать базу данных для книжного магазина BookBy.

В базе данных необходимо хранить данные о клиентах, о продаваемых книгах и об особенностях заказов.

База данных содержит таблицы: Customers (Клиенты), Orders (Заказы) и Books (Книги).

Схема базы данных имеет следующий вид:

Customers (CustomerID, Name, Address, City)
Orders (OrderID, CustomerID, Amount, Date)
Books (ISBN, Author, Title, Price)
Order_ltems (OrderID, ISBN, Quantity)
Book_Reviews (ISBN, Reviews)

Ответ:

```
CREATE DATABASE bookby;
use bookby;

CREATE TABLE customers (
     customerid int unsigned not null auto_increment primary
key,
     name char(50) not null,
```

```
        address char(100) not null,
        city char(30) not null
    );

    CREATE TABLE orders (
        orderid int unsigned not null auto_increment primary key,
        customerid int unsigned not null,
        amount float(6,2),
        dt date not null
    );

    CREATE TABLE books (
        isbn char(13) not null primary key,
        author char(50),
        title char(100),
        price float(4,2)
    );

    CREATE TABLE order_items (
        orderid int unsigned not null,
        isbn char(13) not null,
        quantity tinyint unsigned,
        primary key (orderid, isbn)
    );

    CREATE TABLE book_reviews (
        isbn char(13) not null primary key,
        review text
    );
```

Добавить в базу данных BookBy по 2-3 записи в каждую таблицу используя SQL запросы.

```
    INSERT INTO customers VALUES
          (NULL, "Иванов И.И.", "пр. Пушкина 32", "Минск"),
          (NULL, "Петров И.В.", "пр. Рокоссовского 10", "Минск"),
          (NULL, "Сидоров А.В.", "ул. Цветочная 18", "Пинск");

    INSERT INTO orders VALUES
          (NULL, 3, 69.98, "2006-04-02"),
          (NULL, 1, 49.99, "2006-04-15"),
          (NULL, 2, 74.98, "2006-04-19"),
          (NULL, 3, 24.99, "2006-05-01");

    INSERT INTO books VALUES
          ("0-672-31697-8", "Толстой", "Война и мир", 34.99),
          ("0-672-31745-1",  "Булгаков",  "Мастер  и  Маргарита",
24.99),
          ("0-672-31509-2", "Лермантов", "Избранные произведения",
24.99),
```

```
        ("0-672-31769-9", "Пушкин", "Евгений Онегин", 49.99);

    INSERT INTO order_items VALUES
        (1, "0-672-31697-8", 2),
        (2, "0-672-31769-9", 1),
        (3, "0-672-31769-9", 1),
        (3, "0-672-31509-2", 1),
        (4, "0-672-31745-1", 3);

    INSERT INTO book_reviews VALUES
        ("0-672-31697-8", "Это замечательная книга, ее должен
каждый прочитать.");
```

Задание 1: Создать базу данных «King Hotel» в реляционной СУБД, содержащую следующие таблицы:

Hotel (hotel_no, name, address)
Room (room_no, hotel_no, type, price)
Booking (hotel_no, guest_no, date_from, date_to, room_no)
Guest (guest_no, name, address)

Таблица Hotel содержит сведения о гостинице, причем атрибут hotelno является ее первичным ключом.

Таблица Room содержит данные о номерах всех гостиниц, а комбинация атрибутов (hotelno, room_no) образует ее первичный ключ.

Таблица Booking содержит сведения о бронировании гостиничных номеров: ее первичным ключом является комбинация атрибутов (hotel_no, guest_no, date_from).

Таблица Guest содержит сведения о постояльцах гостиниц, и ее первичным ключом является атрибут guest_no.

Добавить в базу данных «King Hotel» по 2-3 записи в каждую таблицу, используя SQL запросы.

Задание 2: Создать базу данных «Greedy Wallet» в реляционной СУБД.

Данные о клиенте (Customers) включают его имя (name), адрес (address), номер телефона (phone) и код полиса пенсионного страхования (kod_polis).

Счет (Accounts) описывается атрибутами номера (number), типа (например, «накопительный», «чековый» и т. п.) (type) и остатка (balance).

Предусмотреть возможность связи счётов и клиентов типа «многие-ко-многим».

Практическая работа №12 «PostgreSQL»

Данная практическая работа заключается в создании и заполнении отношений БД. Работа выполняются под управлением СУБД PostgreSQL версии не ниже 9.5.

В задании могут использоваться следующие обозначения типов данных:

N – числовой тип (numeric), значение "Дес." при этом указывает количество знаков после запятой (по умолчанию 0);

C – символьный тип (char или varchar в зависимости от длины и семантики поля);

D – тип "дата" (datetime или date в зависимости от семантики поля),
I – тип "интервал" (interval).
Другие обозначения:
ПК – первичный ключ;
ВнК – внешний ключ.

1. Отношение "Кафедры" (поля "Шифр" (ПК), "Название кафедры", "Факультет").
2. Отношение "Дисциплины" (поля "Код дисциплины" (ПК), "Название", "Курс", "Количество лекций", "Количество семинаров", "Количество практических").
3. Отношение "Преподаватели":

Содержимое поля	*Тип*	*Длина*	*Дес.*	*Примечание*
Табельный номер	N	6	0	первичный ключ
ФИО	C	50		обязательное поле
Пол	C	1		значения – 'м' и 'ж', по умолчанию - 'ж'
Дата рождения	D			обязательное поле
Кафедра	C	6		внешний ключ к таблице «Кафедры»
Должность	C	30		ассистент, преподаватель, доцент, профессор
Ученая степень	C	20		кандидат или доктор наук
Ученое звание	C	20		доцент или профессор

4. Отношение "Нагрузка" (поля Id (ПК), "Преподаватель" (ВнК), "Дисциплина" (ВнК), "Тип нагрузки" – лекция, семинар или практические занятия).

Практическая работа №13 «PostgreSQL - 2»

Во данной практической работе производится выборка данных из созданных отношений в предыдущей практической работе. Один из запросов надо написать двумя способами и объяснить, какой из вариантов будет работать быстрее и почему.

Перечень запросов:

- докторов наук с указанием названия кафедры и должности;
- самых молодых преподавателей на каждой кафедре;
- дисциплин и видов нагрузки, для которых не назначены преподаватели;
- кафедр, на которых работает менее 2-х докторов наук.
- для каждой кафедры посчитать количество преподавателей с разными учеными степенями.

В качестве ответа на задание предоставляется отчет в электронном виде, содержащий тексты запросов, пояснения к ним и ответ на вопрос о том, какой вариант запроса будет работать быстрее.

Практическая работа №14 «PostgreSQL - 3»

Создать:
1. Представление "Ассистенты, читающие лекции".

2. Представление "Распределение дисциплин": дисциплина – количество лекций – ФИО лектора – количество семинаров – ФИО преподавателя – количество практических – ФИО лаборанта. Если количество семинаров или практических равно нулю, то выводится NULL.
3. Представление "Нагрузка преподавателей": ФИО преподавателя – дисциплина – количество лекционных часов (количество лекций, умноженное на 2) – количество семинаров – количество практических – общее количество часов по дисциплине.

Для созданных представлений необходимо проверить с помощью запросов UPDATE и INSERT, являются ли они обновляемыми, и объяснить полученный результат.

В качестве ответа на задание предоставляется отчет в электронном виде, содержащий тексты запросов, пояснения к ним и скриншоты их выполнения.

Задания для самостоятельной работы студентов:

1. Создать проект базы данных «Фирма», представленный отношениями:

- Филиалы (Филиал, Страна_Филиала),
- Заказы (Филиал, Заказчик, № Заказа),
- Товары (№ заказа, Товар, Количество, Цена),
- Клиенты (Заказчик, Страна_Заказчика).

Обозначить ключевые поля, поля связи, тип связи.

2. Создать проект базы данных «Банк», представленный отношениями:

- Клиент (ФИО клиента, № филиала, № счета, Остаток, Кредит),
- Филиал (№ Филиала, Район)

Обозначить ключевые поля, поля связи, тип связи.

3. Создать проект базы данных «Программные комплексы», представленный отношениями:

- Файлы (Файл, Сотрудник),
- ПК_Файлы (Код_ПК, Файл),
- Программные комплексы (Код_ПК, Название ПК),
- Сотрудники (Сотрудник, Отдел)

Обозначить ключевые поля, поля связи, тип связи.

4. Открыть БД Банк. Составить запрос: Клиентов, которые имеют счета с остатком, превышающим 10 000 руб.

5. Открыть БД Банк. Составить запрос: Клиентов, которые имеют по одному счету

6. Открыть БД Банк. Составить запрос: Выбрать клиентов, имеющих на счете до 200руб.

7. Открыть БД Банк. Составить запрос: Выбрать клиентов, имеющих на счете 0 руб.

8. Открыть БД Банк. Составить запрос: Выбрать клиентов из Салавата и Ишимбая

9. Открыть БД Банк. Составить запрос: Выбрать клиентов, у которых сумма кредита больше суммы остатка

10. Открыть БД Банк. Составить запрос: Клиентов, у которых сумма кредита не превышает сумму остатка более чем в два раза

11. Открыть БД Банк. Составить запрос: Клиентов, которые имеют на счету остаток ноль.

12. Открыть БД Банк. Составить запрос: Клиентов, которые имеют несколько счетов

13. Открыть БД Банк. Составить запрос: Филиалы с общей суммой выданного кредита

14. Открыть БД Банк. Составить запрос: Клиентов, у которых сумма кредита не превышает 50% от суммы остатка

15. Открыть БД Банк. Составить запрос: Клиентов, обслуживающихся в филиале №3

16. Открыть БД Банк. Составить запрос: Клиентов с общей суммой полученного кредита

17. Открыть БД «Сотрудники». Составить запрос: «Должности, занимаемые сотрудниками». Поле «ФИО» в запросе отсортировать по возрастанию

18. Открыть БД «Сотрудники». Составить запрос: «Работающие, свыше 7 лет», поле - стаж - не выводить на экран

19. Открыть БД «Сотрудники». Составить запрос: «Сложность работ» (сложность выполняемых работ – по возрастанию)

20. Открыть БД «Сотрудники». Составить запрос: «Заработная плата сотрудников». Поле «Зарплата» отсортировать по убыванию. Вывести в запросе поля: ФИО, Должность, Зарплата. В запросе не выводить поля, из которых формируется заработная плата

21. Открыть БД «Сотрудники». Составить запрос: «Работающие в должности…» (запрос с параметром). Вывести в запросе поля: ФИО, Должность.

22. Открыть БД «Сотрудники». Составить запрос: «Сотрудники – мужчины 1978 и 1979 г. рождения». Вывести в запросе поля: ФИО, Должность. Поле «дата рождения» - не выводить.

23. Открыть БД «Сотрудники». Составить запрос: «Средняя заработная плата»

24. Открыть БД «Сотрудники». Составить запрос: «Минимальная заработная плата»

Пример проектного задания для студентов:

Необходимо разработать следующие SQL-скрипты:

1. SQL-скрипт для создания таблиц БД и связей между ними, который проверяет наличие всех таблиц и объектов на их существование, если они есть, удаляет их из системной базы данных и создает заново.
2. SQL-скрипт, создающий хранимые процедуры для добавления, удаления и редактирования каждой из таблиц. Каждая такая хранимая процедура должна обеспечивать проверку на обеспечение целостности данных.
3. SQL-скрипт для наполнения БД тестовыми значениями. Для вставки значений в таблицу необходимо использовать ранее разработанные хранимые процедуры, поэтому запрещается использовать операторы INSERT.
4. SQL-скрипт, производящий очистку всех таблиц с последующим их удалением.
5. SQL-скрипт, содержащий набор хранимых процедур (этот набор можно поместить в SQL-скрипт №2), обеспечивающих все необходимые функции обработки данных, предусмотренных вариантом задания, а именно:
   - Запрос «Исходные данные» - выдает все данные из всех таблиц, представляя их в удобной для восприятия форме при этом исключить повторение первичных ключевых полей из главных таблиц;
   - Запрос "Проживающие по заданному адресу", который выдает список проживающих на заданной улице;
   - Запрос "Алфавитный список", который выдает список фамилий в указанном буквенном диапазоне. Начальную и конечную буквы диапазона выбрать самостоятельно. Отсортировать список в алфавитном порядке.
   - Запрос "Список с условием" – согласно варианту задания (см. ниже).
   - Запрос "Фамилия, преобразованная к верхнему регистру", который выдает список всех фамилий, преобразованных к верхнему регистру.
   - Запрос "Фамилии и инициалы", который выдает список всех фамилий и инициалов. В запрос включить поля: Фамилия, Имя, Отчество (для контроля выполнения запроса).
   - Запрос с расчетами – согласно варианту задания (см. ниже).
   - Запрос с параметрами – согласно варианту задания (см. ниже).
   - Запрос с групповой операцией – согласно варианту задания (см. ниже).
   - Запрос с групповой операцией SUM – согласно варианту задания (см. ниже).
   - Запрос с групповой операцией COUNT – согласно варианту задания (см. ниже).

Пример варианта: (5)

База данных "Библиотека" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество читателя.
2. Домашний адрес читателя.
3. Телефон читателя.
4. Дата рождения читателя.
5. Номер читательского билета.
6. Автор книги, которую взял читатель.
7. Название книги.
8. Дата выдачи книги.
9. Дата возврата книги.

10. Цена книги.

Запрос «Список с условием»: список книг, цена которых не превышает определенного значения. Значением цены книги задаться самостоятельно.

Запрос «Запрос с расчётами»: список, отсортированный по номеру билета, содержащий:

- Фамилия
- Имя
- Отчество
- Номер билета
- Название книги
- Цена книги
- Цена книги, увеличенная на 30%

Запрос «с параметрами»: Список читателей, пользовавшихся указанной книгой, и стоимость этой книги (параметр: название книги). Формат вывода стоимости: пользовательский, 2 знака после запятой.

Запрос «с групповой операцией»: вывести среднюю цену книг

Запрос с групповой операцией SUM: вывести суммарную стоимость книг, выданных каждому читателю.

Запрос с групповой операцией COUNT: вывести количество книг, выданных каждому читателю.

Критерии оценки:
Максимально можно получить 14 баллов.
За каждый скрипт можно получить от 0 до 2 баллов, в зависимости от правильности выполнения. Всего 5 * 2 = 10 баллов.
От 0 до 4 баллов можно получить за полноту и корректность отчёта и выступления по проделанной работе.