

Лабораторная работа №21

Тема работы: «Исследование возможностей среды Workbench. Создание объектов БД»

1. Цель работы

Научиться применять знания о создании объектов баз данных; исследовать возможности среды Workbench.

2. Задание

1 Создать базу данных, показанную в примере. Организовать связи в данной базе и осуществить загрузку базы на сервер.

Для примера рассмотрим базу данных «службы доставки». Данная база будет состоять из 7 таблиц.

Таблица Пользователи	Таблица Настройки	Таблица Магазины	Таблица Заказы
Код пользователя	Код настройки	Код магазина	Код заказа
ФИО	Адрес сервера	Имя магазина	Дата
Логин	Имя базы данных	Адрес	Количество
Пароль	Имя пользователя	Телефон	Инф. о подтверждении заказа
Е-mail	Пароль	Адрес сайта	Телефон
Тип пользователя		Е-mail	

Таблица Продукты	Таблица Тип продукта	Таблица Доставки	Таблица Состав заказа
Код продукта	Код тип продукта	Код доставки	Код состава заказа
Имя магазина	Имя продукта	Адрес	
Бренд		Время	
Модель		Дата	
Характеристики продукта		Инф. о доставке	
Адрес до изображения продукта			
Стоимость			
Инф. о сроке гарантии			

2 Создать базу данных согласно варианту индивидуального задания.

- Организовать связи в созданной БД.
- Осуществить загрузку БД на сервер.
- Осуществить вывод информации из созданной БД по усмотрению преподавателя.

Варианты заданий

Вариант 1 База данных «Деканат»

Таблица Студенты	Таблица Преподаватели	Таблица Дисциплины	Таблица Экзамены
Код студента	Код преподавателя	Код дисциплины	Код экзамена
Имя	Имя	Наименование	Код студента
Фамилия	Фамилия	Кафедра	Код дисциплины
Отчество	Отчество	Специальность	Код преподавателя
Дата Рождения	Факультет	Код преподавателя	Оценка

Группа
Факультет
Специальность

Должность

Экзамен N семестра
Зачет N семестра

Семестр

Вариант 2 База данных «Товары магазина бытовой техники»

Таблица Товары
Код товара
Наименование
Стоимость
Наличие
Количество
Характеристики
Код заказа

Таблица Заказы
Код заказа
Код товара
Дата размещения
Дата исполнения
Код клиента
Код сотрудника

Таблица Клиенты
Код клиента
ФИО
Адрес
Телефон

Таблица Сотрудники
Код сотрудника
Имя
Фамилия
Отчество
Должность
Адрес

Вариант 3 База данных «Ресторан»

Таблица Заказы
Код заказа
Дата заказа
Номер столика
Код официанта
Сумма

Таблица Официанты
Код официанта
Имя
Фамилия
Отчество

Таблица Блюда
Код блюда
Название
Количество
Состав

Таблица Заказ из меню
Код официанта
Код заказа
Код блюда
Количество
Сумма

Вариант 4 База данных «Выпускники учебного заведения»

Таблица Выпускник
Код выпускника
ФИО
Год поступления
Год выпуска
Код учебного заведения
Код организации
Адрес
Телефон

Таблица Учебное заведение
Код учебного заведения
Наименование
Направление учебного заведения
Дата поступления

Таблица ЦТ
Код выпускника
Название предмета
Количество баллов
Преподаватель

Таблица Трудоустройство
Код организации
Наименование
Должность
Дата
Контактная информация

Вариант 5 База данных «Оказания услуг салона красоты»

Таблица Услуги
Код услуги
Название
Цена
Код сотрудника
Описание услуги

Таблица Посещения
Код посещения
Код клиента
Код услуги
Код сотрудника
Дата

Таблица Клиенты
Код клиента
ФИО
Телефон
Адрес
Отметка о постоянстве посещения

Таблица Сотрудники
Код сотрудника
Фамилия
Имя
Отчество
Должность

Время
Отметка о выполнении услуги

Телефон
Адрес

Вариант 6 База данных «Наличие лекарств в аптеках»

Таблица Лекарства

Код лекарства
Наименование
Код производителя
Форма отпуска
Дозировка
Годен до

Таблица Наличие

Код наличия
Код лекарства
Количество
Цена
Код аптеки

Таблица Аптеки

Код аптеки
Название
Адрес
Телефон
Часы работы

Таблица Производители

Код производителя
Название
Страна

Вариант 7 База данных «Химчистки»

Таблица Услуги

Код услуги
Филиал
Вид услуги
Код клиента
Дата приема
Код срочности
Код сложности
Дата возврата

Таблица Клиенты

Код клиента
Имя
Фамилия
Отчество
Отметка о постоянстве
Код срочности
Код сложности

Таблица Срочность

Код срочности
Название
Надбавка

Таблица Сложность

Код сложности
Название
Надбавка

Вариант 8 База данных «Учет постояльцев в гостинице»

Таблица Клиенты

Код клиента
Код номера
Код сотрудника
Имя
Фамилия
Отчество
Дата Рождения
Пол
Дата заезда
Дата выезда
Сумма
Код брони

Таблица Сотрудники

Код сотрудника
Имя
Фамилия
Отчество
Дата рождения
Должность

Таблица Номера

Код номера
Категория
Количество мест
Доп информация
Стоимость в сутки

Таблица Бронь

Код брони
Код клиента
Список фамилий

Вариант 9 База данных «Кинотеатра»

Таблица Билеты

Код билета
Дата
Код сеанса
Место
Оплачено

Таблица Залы

Код Зала
Наименование зала
Вместимость
Описание
Число мест

Таблица Фильмы

Код фильма
Название фильма
Производство
Год выпуска
Жанр

Таблица Сеансы

Код сеанса
Дата
Время
Код Зала
Свободные места

Длительность

Код фильма

Вариант 10 База данных «Учета читателей в библиотеке»

Таблица Книги	Таблица Читатели	Таблица Выдача книг	Таблица Штрафы
Код книги	Код читателя	Код выдачи	Код штрафа
Название	Фамилия	Код книги	Код выдачи
Автор	Имя	Код читателя	Штрафная статья
Жанр	Отчество	Дата выдачи	
Стоимость	Адрес	Ожидаемая дата возврата	
	Телефон	Фактическая дата возврата	
		Сумма штрафов	

Вариант 11 База данных «Проката автомобилей»

Таблица Выданные автомобили	Таблица Автомобили	Таблица Клиенты	Таблица Денежные взыскания
Код проката	Код авто	Код клиента	Код штрафа
Код клиента	Госномер	Фамилия	Код проката
Код_авто	Марка	Имя	Наименование штрафа
Дата выдачи	Модель	Отчество	Сумма штрафа
Имя директора	Стоимость	Адрес	Наименование скидки
Количество дней	Стоимость 1 суток проката	Телефон	Процент
Ожидаемая дата возврата			
Сумма			
Скидка			
Штрафы			
Итоговая сумма			

Вариант 12 База данных «Ювелирной мастерской»

Таблица Продажи	Таблица Покупатели	Таблица Материалы	Таблица Изделия
Код продажи	Код покупателя	Код материала	Код изделия
Код изделия	Фамилия	Материал	Тип
Дата продажи	Имя	Код изделия	Название
Код_покупателя	Отчество	Количество	Стоимость материала
Скидка	Телефон	Стоимость материала	Стоимость изготовления
Цена со скидкой	Адрес		Цена

Вариант 13 База данных «Парикмахерской»

Таблица Филиалы	Таблица Клиенты	Таблица Посещения	Таблица Услуги
-----------------	-----------------	-------------------	----------------

Код филиала	Код клиента	№ по порядку	№ по порядку
Адрес	ФИО	Код филиала	№ посещения
Название	Телефон	Дата	Название услуги
Телефон	Дата рождения	Код клиента	Пол
Управляющий	Пол	Отметка о постоянстве	
		Сумма для оплаты	

3. Оснащение работы

Технические средства обучения:

- IBM – совместимый компьютер;

Электронные средства обучения:

- веб-браузер (Opera, Google Chrome, Mozilla Firefox и др.);
- текстовый редактор MySQL Workbench.

4. Основные теоретические сведения

MySQL Workbench — это графический инструмент для работы с базами данных MySQL. MySQL — это система управления реляционными базами данных с открытым исходным кодом, и это самая популярная система баз данных, используемая с PHP.

MySQL Workbench — мощный визуальный инструмент для администраторов баз данных, архитекторов баз данных и разработчиков MySQL. Этот инструмент предлагает моделирование данных, разработку SQL и инструменты администрирования для настройки и администрирования сервера.

MySQL Workbench предоставляет возможности для управления моделями баз данных, такими как:

- Создание графической модели
- Обратный инжиниринг живых баз данных в модели (моделирование данных)
- Прямая инженерная модель в скрипт / живую базу данных и больше

Моделирование и дизайн

Моделирование — отличный способ визуализировать требования и создавать хорошо работающие базы данных, которые могут не отставать от постоянно меняющихся требований к данным. MySQL Workbench позволяет создавать модели и управлять ими, преобразовывать динамическую базу данных в модель, а также создавать и редактировать таблицы и вставлять данные.

Вы можете преобразовать диаграммы ER в операторы SQL и отправить их на сервер SQL. Вы также можете создавать модели из целевой базы данных или даже из импортированных файлов SQL.

Инструмент разработки SQL

MySQL Workbench позволяет создавать, управлять и настраивать подключения и параметры подключения к серверам баз данных MySQL. Он также позволяет выполнять SQL-запросы к этим соединениям с помощью встроенного редактора.

Редактор Visual SQL позволяет создавать, редактировать и выполнять запросы. Он имеет автозаполнение и цветные выделения, которые помогают легко писать и отлаживать операторы SQL.

Визуальное представление

MySQL Workbench предлагает различные инструменты, которые позволяют просматривать и улучшать производительность. Панель управления производительностью дает вам быстрый обзор различных показателей производительности. Отчеты о производительности позволяют анализировать производительность ваших баз данных.

Перенос базы данных

MySQL Workbench упрощает процесс миграции. Он позволяет выполнять миграцию с Microsoft SQL Server, SQLite, Microsoft Access и многих других. Он также позволяет настраивать, планировать, редактировать и выполнять проекты миграции.

Управление подключением

MySQL Workbench позволяет создавать, управлять и организовывать подключения к базе данных. Управление объектами Обзорщик объектов MySQL Workbench позволяет визуально выбирать таблицы и столбцы. Вы можете легко перемещаться между схемами, выбирать таблицы и поля, создавать новые или отбрасывать их.

Конфигурация сервера

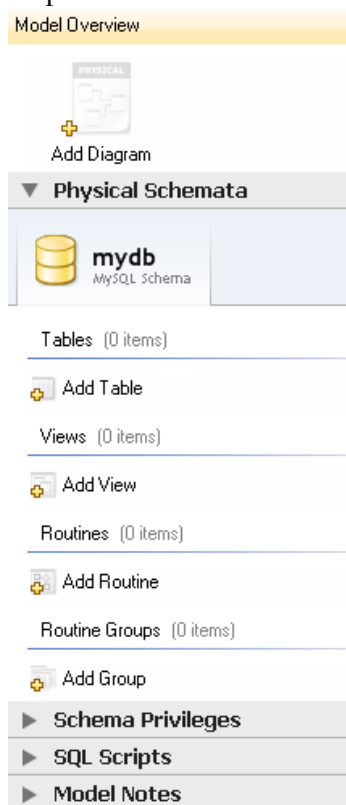
MySQL Workbench позволяет точно настраивать серверы MySQL, позволяя просматривать и редактировать расширенные параметры. Вы также можете просматривать журналы сервера, чтобы быстрее выявлять проблемы и отслеживать изменения в базе данных. Вы можете указать имя хоста MySQL, чтобы определить, где разместить вашу базу данных MySQL.

Экспорт и импорт

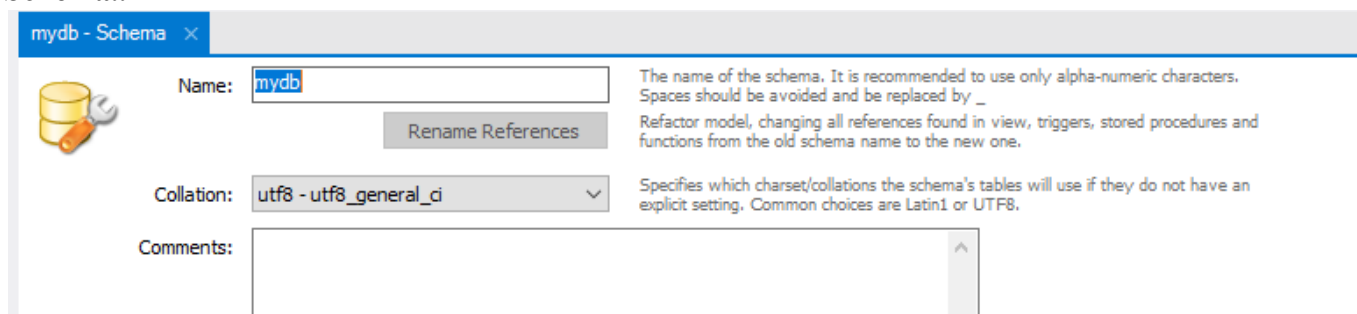
MySQL Workbench позволяет легко экспортировать или импортировать данные, выбирая схемы или таблицы базы данных в обзорщике объектов.

Порядок создания БД средствами Workbench

1. Для создания новой модели, воспользуемся ссылкой Create New EER Model, после нажатия на неё отобразится окно с параметрами:



2. Далее можно изменить имя модели, нажав правой кнопкой мыши по **mydb** и выбрав **Edit Schema..**



3. Далее необходимо создать таблицы, для этого нажмём на кнопку Add Table, появится следующая форма:

table1 - Table ×

Table Name: Schema: **mydb**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
		☐	☐	☐	☐	☐	☐	☐	☐	

Сначала создадим таблицу **users**, которая будет хранить данные о пользователях информационной системы, в поле **table Name** впишем имя таблицы, в разделе формы **Columns** создадим поля таблицы:

– Первое поле **id** будет содержать уникальный номер пользователя, зададим ему свойства: **Auto Increment**, **Not Null**, **Primary key** и **Unique**, в разделе **Data type** выберем целочисленный тип **integer (INT)**.

– Второе поле **fio**, где будет храниться **Ф.И.О.** пользователя, установим полю свойства: **Not Null**, в разделе **Data type** выберем строковый тип **VARCHAR** и зададим количество символов в **255**.

– Третье поле **login**, будет содержать логин пользователя, оно должно быть уникальным, как и поле **id**, поэтому установим ему свойство **Unique** и зададим количество символов в **255**.

– Следующие поля: **password** содержащее пароль, **e_mail** содержащее адрес электронной почты и поле **type** содержащее тип пользователя будут без особых свойств, со строковым типом **VARCHAR** длиной в **255** символов, за исключением последнего поля **type** которому хватит с **45** символов.

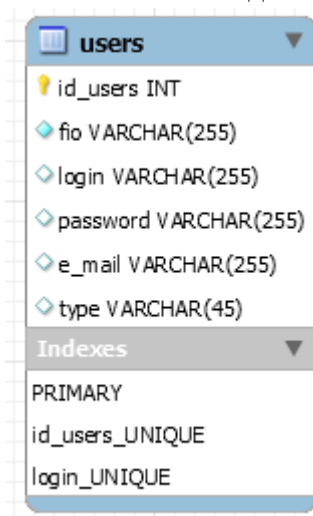
После проделанных манипуляций форма с именем таблицы **users** будет выглядеть так:

users - Table ×

Table Name: Schema: **mydb**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_users	INT	☒	☒	☒	☐	☐	☐	☒	☐	
fio	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
login	VARCHAR(255)	☐	☐	☒	☐	☐	☐	☐	☐	
password	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
e_mail	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
type	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	

На диаграмме появится таблица **users** с полями и индексами:



4. Аналогичным способом создадим таблицу **settings** с настройками доступа к базе данных ИС, содержащую поля **id_settings**, **host** для указания имени хоста (адреса сервера), **db** – имени

базы данных, **user** и **password** с именем пользователя и паролем, для установки ИС на удалённый сервер.

settings - Table ×										
		Table Name: settings							Schema: mydb	
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 id_settings	INT	☒	☒	☒	☐	☐	☐	☒	☐	
 host	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	
 db	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	
 user	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	
 password	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	

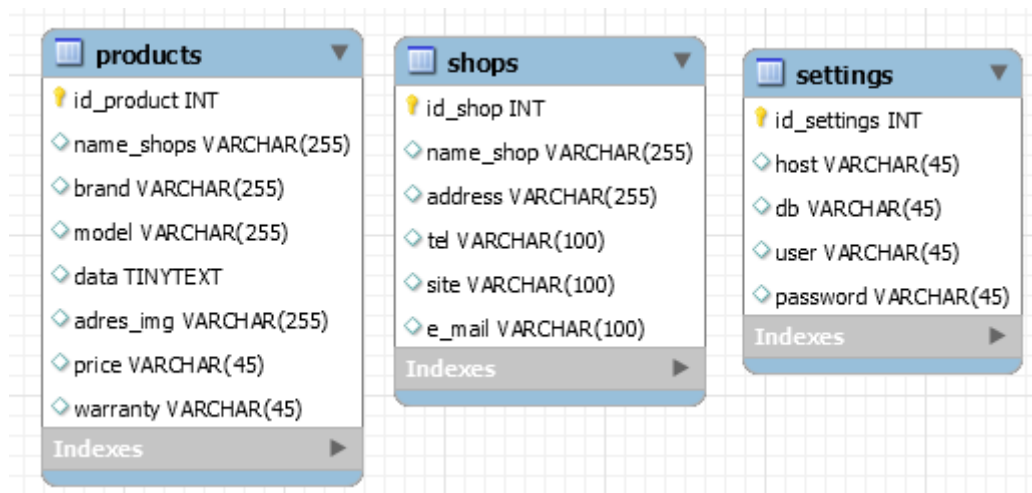
Далее по уже известному методу создадим таблицу **shops** которая будет хранить данные о магазинах в полях: **id_shop** типа **integer** – ключевое, ненулевое, уникальное с автоинкрементом, поле **name_shop** хранящее название магазина, поле **address** – его физический адрес, поле **tel** – телефон магазина, **site** – интернет сайт магазина и поле **email** с электронным адресом магазина.

shops - Table ×										
		Table Name: shops							Schema: mydb	
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 id_shop	INT	☒	☒	☒	☐	☐	☐	☒	☐	
 name_shop	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 address	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 tel	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	
 site	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	
 e_mail	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	

Затем создадим таблицу **products** хранящую данные о продукции магазина в полях: **id_product** типа **integer** – ключевое, ненулевое, уникальное с автоинкрементом, поле **name_shops** хранящее название магазина. Поле **brand** – бренд изготовителя длиной в 255 символов, поле **model** – с моделью товара, поле **data** – с данными и характеристиками товара типа **Tinytext**, поле **adres_img** с полным адресом до изображения товара длиной в 255 символов, и поля **price** с ценой товара и **warranty** с информацией о сроках гарантии на товар длиной в 45 символов.

products - Table ×										
		Table Name: products							Schema: mydb	
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 id_product	INT	☒	☒	☒	☐	☐	☐	☒	☐	
 name_shops	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 brand	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 model	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 data	TINYTEXT	☐	☐	☐	☐	☐	☐	☐	☐	
 adres_img	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
 price	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	
 warranty	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	

Созданные таблицы **settings**, **shops** и **products** выглядят следующим образом:



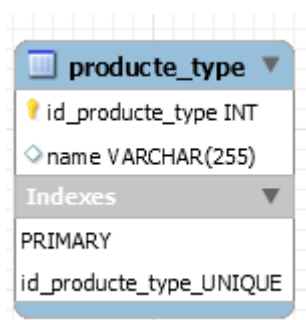
5. Далее нам понадобится таблица, хранящая тип продуктов **product_type**, она состоит из уникального, ненулевого, ключевого поля **id_product_type** с автоинкрементом целочисленного типа, и уникального поля **name** длиной в 255 символов, которое содержит название вида продуктов.

producte_type - Table

Table Name: Schema: **mydb**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<code>id_producte_type</code>	INT	☒	☒	☒	☐	☐	☐	☒	☐	
<code>name_product</code>	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	

Вид таблицы таков:



6. Еще две таблицы это **orders** и **deliveries**, первая содержит информацию о заказах клиентов, а последняя данные о доставке продукции.

Поля таблицы **orders**: **id_orders** ключевое, ненулевое, уникальное поле целочисленного типа с автоинкрементом, поле **date** с датой заказа — типа **DATE**, поле **quantity** с количеством заказанных товаров — целочисленного типа, поле **tel** с номером телефона заказчика — строкового типа длиной в 255 символов и поле **confirm** содержащее информацию о подтверждении заказа — логического типа.

orders - Table

Table Name: Schema: **mydb**

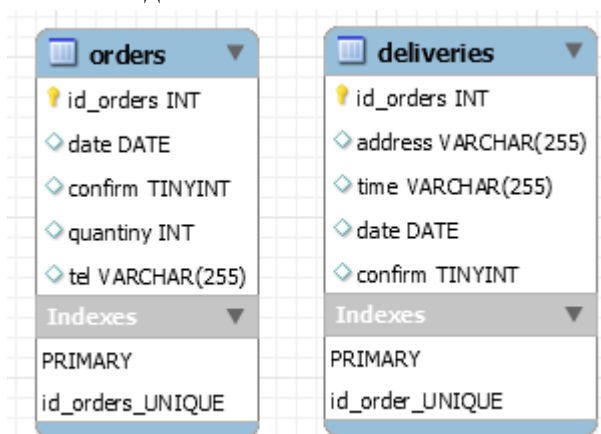
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<code>id_orders</code>	INT	☒	☒	☒	☐	☐	☐	☒	☐	
<code>date</code>	DATE	☐	☐	☐	☐	☐	☐	☐	☐	
<code>confirm</code>	TINYINT	☐	☐	☐	☐	☐	☐	☐	☐	
<code>quantiny</code>	INT	☐	☐	☐	☐	☐	☐	☐	☐	
<code>tel</code>	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	

Поля таблицы **deliveries**: **id_orders** с номером заказа — ключевое, ненулевое, уникальное поле целочисленного типа с автоинкрементом, поле **id_users** с номером пользователя

совершившего заказ —ненулевое целочисленного типа, поле **address** хранящее адрес доставки товара указанный клиентом – строкового типа длиной в 255 символов, поле **time** хранящее желаемое время доставки товара – строкового типа длиной в 255 символов, поле **date** с датой совершения заказа клиентом – типа **DATE** и поле логического типа **confirm** хранящее информацию о доставке товара.

deliveries - Table										
Table Name: deliveries		Schema: mydb								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_orders	INT	☒	☒	☒	☐	☐	☐	☒	☐	
address	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
time	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
date	DATE	☐	☐	☐	☐	☐	☐	☐	☐	
confirm	TINYINT	☐	☐	☐	☐	☐	☐	☐	☐	

Таблицы **orders** и **deliveries** выглядят так:



6. Последняя таблица это **list_orders**, которая содержит информацию о списках заказов клиентов.

Поля таблицы **list_orders**: **id_list_orders** ключевое, ненулевое, уникальное поле целочисленного типа с автоинкрементом.

list_order - Table										
Table Name: list_order		Schema: mydb								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_list_order	INT	☒	☒	☒	☐	☐	☐	☒	☐	

Таблицы **list_order** выглядит так:

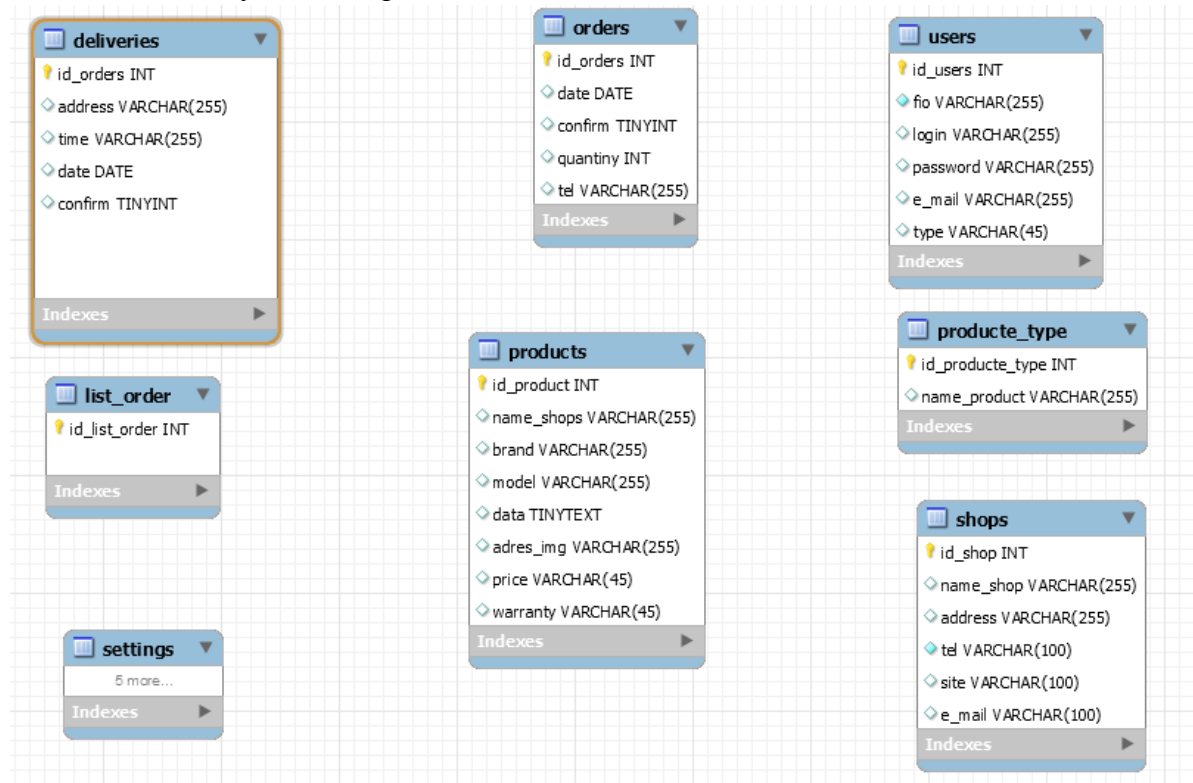


Связи таблиц

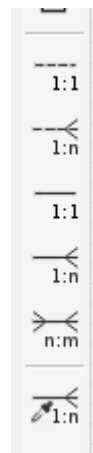
После создания базы данных, состоящую из семи таблиц, теперь необходимо связать

таблицы. Для этого нажимаем значок . В появившемся окне добавляем созданные

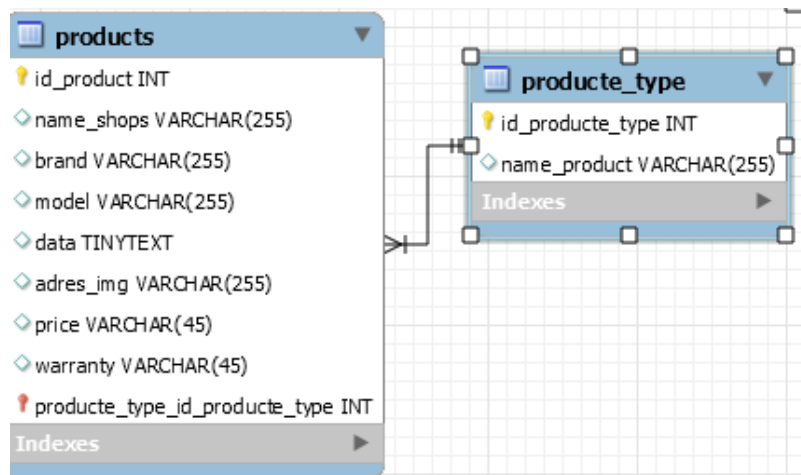
таблицы, если они не добавились автоматически. Для этого в дереве каталога выбираем созданную БД, далее **tables** и выбрав таблицы перетягиваем их в пустое окно. После этого окно диаграммы должно выглядеть следующим образом.



Связи в таблицах осуществляем при помощи следующих инструментов, расположенных слева от главного окна.



Выбираем необходимую связь для каждой из таблицы. Выбираем, например, таблицу **Product** нажимаем на нее ЛКМ и затем на таблицу с которой хотим связать ее, например, **Product_type**. В итоге должно получиться следующим образом.

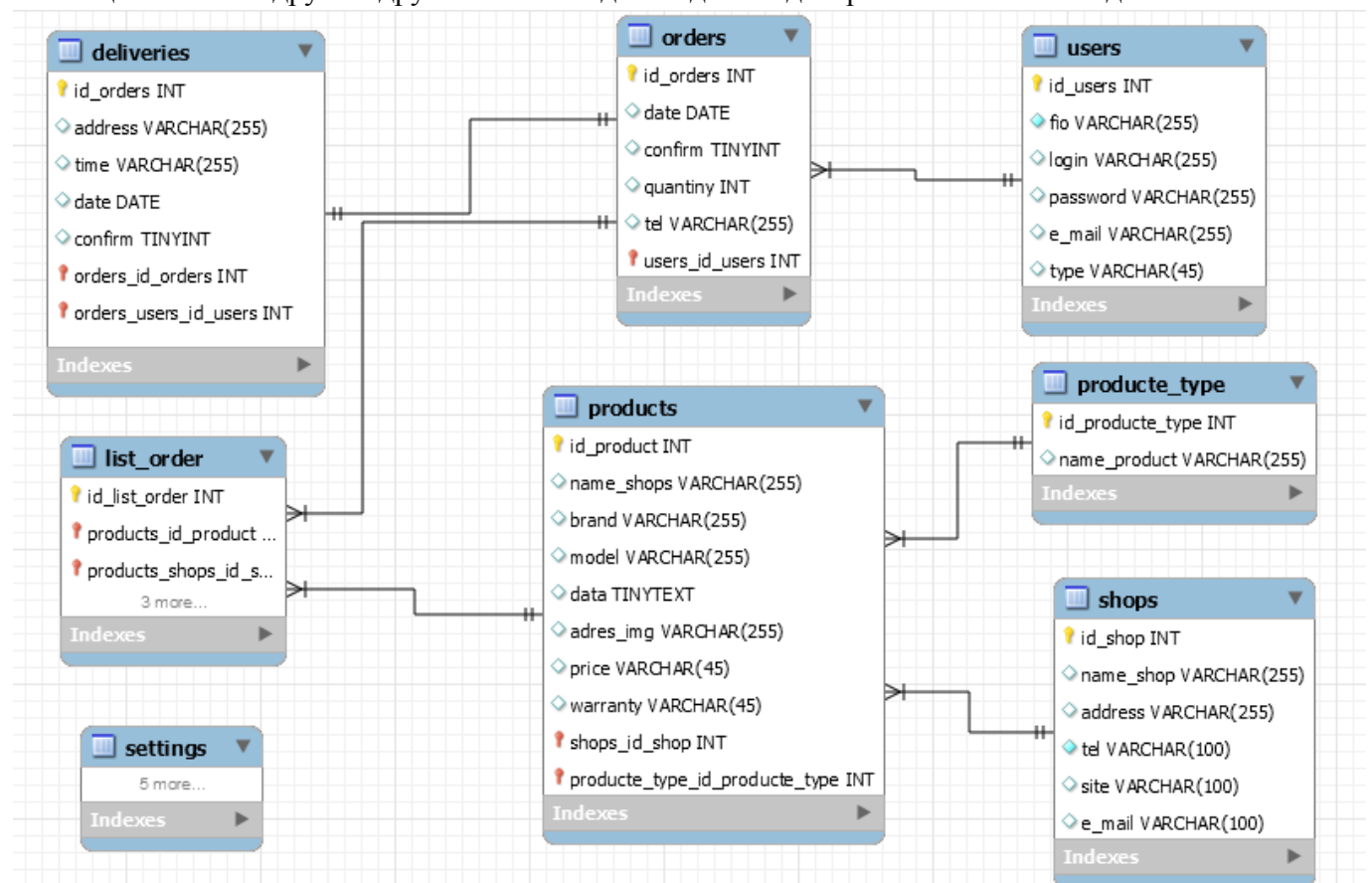


Аналогичным образом связываем все ключевые поля в таблицах, чтобы они были логически взаимосвязаны, далее нам необходимо убедиться, что спроектированная база данных соответствует третьей нормальной форме.

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение. Что же касается различных таблиц, то они могут не быть правильными представлениями отношений и, соответственно, могут не находиться в первой нормальной форме. Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый не ключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа. База данных будет находиться в третьей нормальной форме, если она приведена ко второй нормальной форме и каждый не ключевой столбец независим друг от друга.

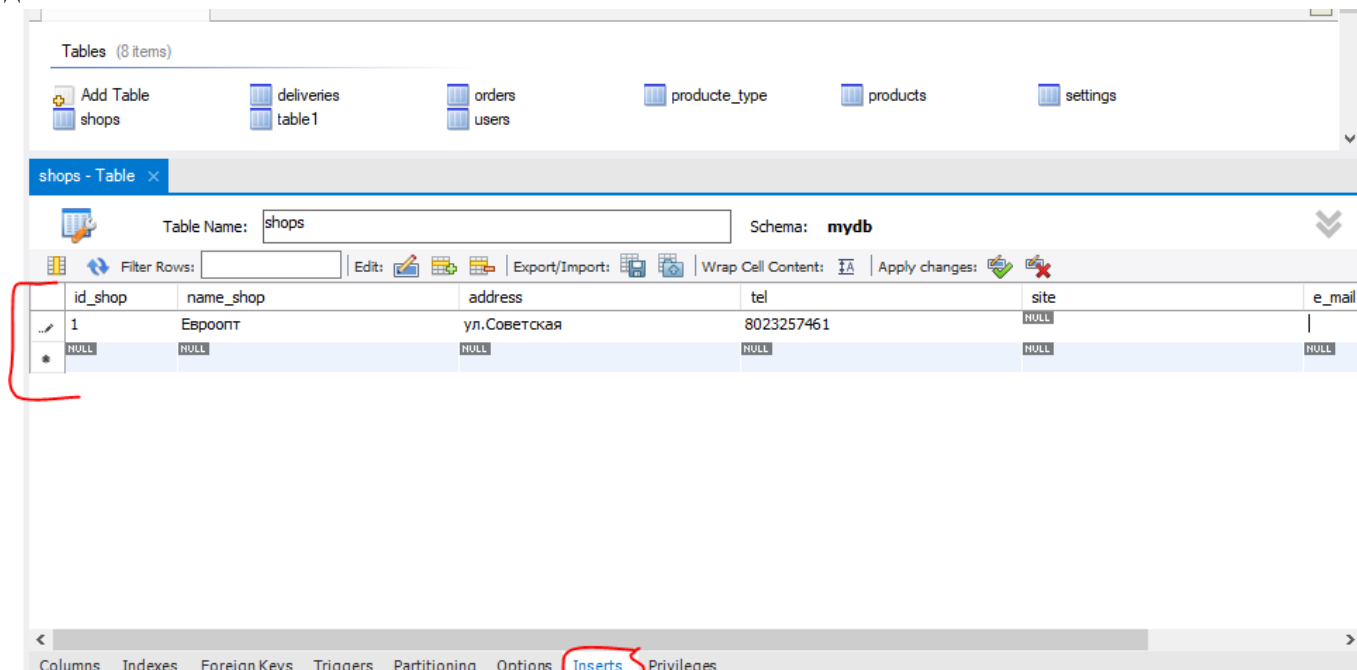
Таким образом база находится в третьей нормальной форме, т.к. каждый не ключевой столбец независим друг от друга. Это наглядно видно на диаграмме нашей базы данных:



Большинство таблиц находятся в отношении один-ко-многим, за исключением таблиц **deliveries** и **orders** находящихся в отношении один-к-одному, т.к. доставлен, может быть только один заказ, т.е. у одного заказа только одна доставка. Остальные связи наглядно указаны выше.

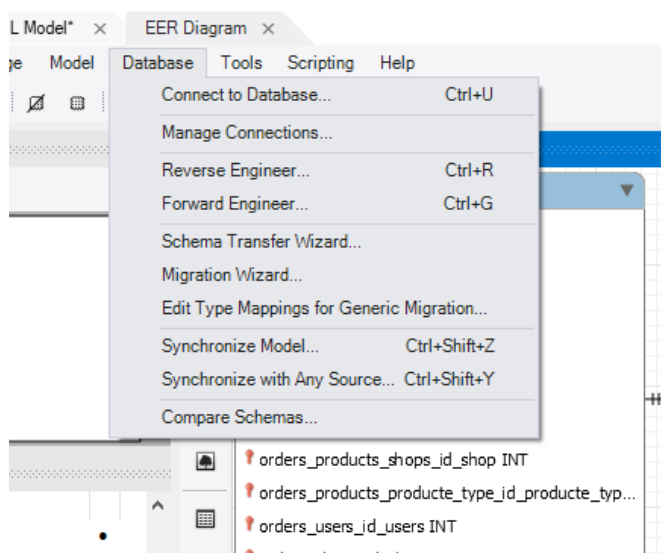
Заполнение базы данных

Для заполнения таблиц данным вернитесь во вкладку **MuSQL Model**. Выберите таблицу, которую нужно заполнить данными и перейдите во вкладку **Inserts**. Заполните появившееся окно данными.



Загрузка БД на сервер

Теперь загрузим базу данных на сервер. Откроем вкладку **EER Diagram**, в панели выберем пункт **Database** и нажмём на параметр **Forward Engineer**:



После того как появится окно, нажимаем на кнопку **“Next”**, выбираем параметр **Export MySQL Table Objects** и нажимаем на кнопку **“Next”**:

Forward Engineer to Database

Connection Options

- Options
- Select Objects
- Review SQL Script
- Commit Progress

Set Parameters for Connecting to a DBMS

Stored Connection: Select from saved connection settings

Connection Method: Method to use to connect to the RDBMS

Parameters **SSL** Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

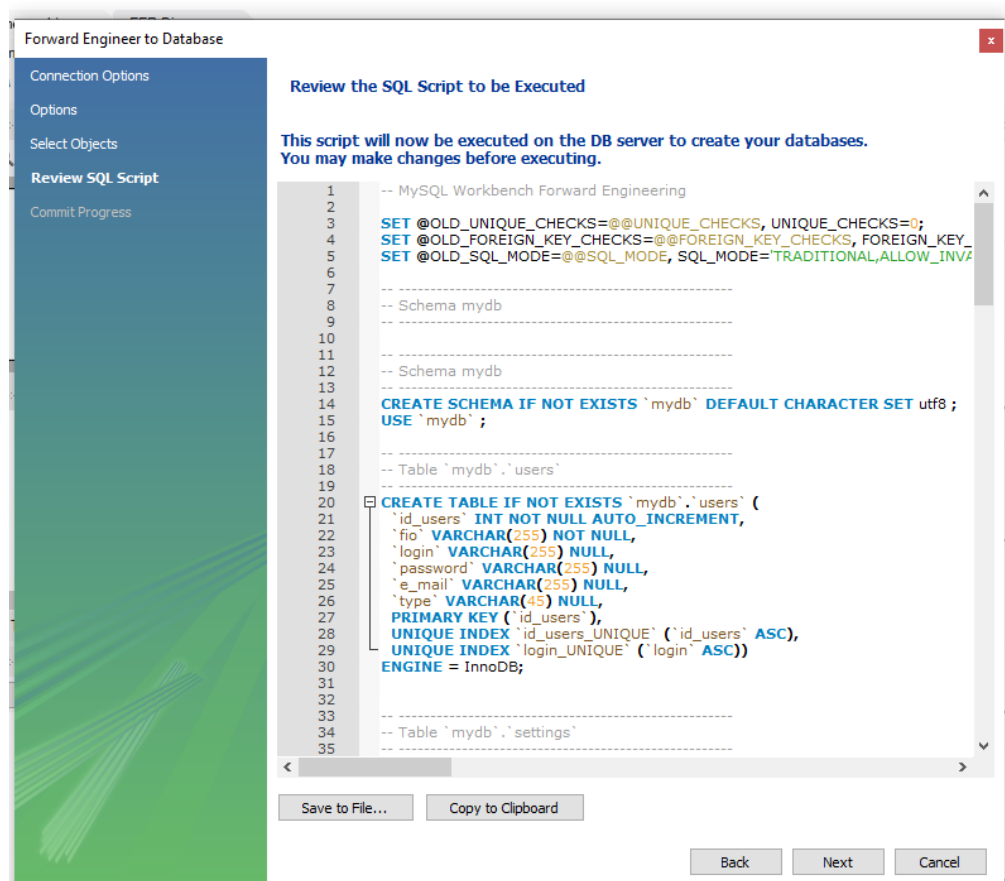
To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.



☒ Export MySQL Table Objects

8 Total Objects, 8 Selected

После нажатия кнопки появится вкладка с SQL кодом, можно сохранить его нажав кнопку **“Save to file”** если это необходимо, а затем нажать на кнопку **“Next”**. Появится окно с параметрами соединения:



Проверяем, верны ли параметры подключения и нажимаем на кнопку “Execute”, если в SQL коде не содержится ошибок, то после исполнения кода мы увидим окно со списком таблиц, иначе выведется сообщение об ошибке. Теперь наша база загружена на сервер.

5. Форма отчета о работе

Номер учебной группы _____
 Фамилия, инициалы обучающегося _____
 Дата выполнения работы _____
 Тема работы: _____
 Цель работы: _____
 Задание: _____
 Оснащение работы: _____
 Результат выполнения работы: _____

6. Контрольные вопросы и задания

1. Описать возможности среды MySQL Workbench.
2. Описать порядок создания БД в среде MySQL Workbench.
3. Объяснить предназначение связей в БД. Назвать виды связей БД, дать им определения.

Рекомендуемая литература

1. Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. М.: Вильямс, 2018. 1382 с.
2. Лазицкас, Е.А. Базы данных и системы управления базами данных / Е.А. Лазицкас, И.Н. Загуменникова, П.Г. Гилевский. Минск : РИПО, 2016. 268 с.
3. Федорова, Г. Разработка и администрирование баз данных / Г. Федорова. М.: Академия, 2015. 313 с.
4. Видеоуроки по MySQL 8.0 https://www.youtube.com/watch?v=EMOKs8_l2WA