

# Базы данных.

## Лабораторная работа №2

Цель лабораторной работы — продолжить знакомство с MySQL Workbench, языком запросов SQL и начать осваивать инструмент проектирования баз данных в визуальном редакторе, который предоставляет это ПО.

Автор: Чалапко Егор Витальевич, студент 2 курса ИВТ, 1 группы, 1 подгруппы

# Задание 1

**Задание 1.** Повторите действия, демонстрируемые в ролике и создайте диаграмму (модель), созданную во второй половине ролика. Экспортируйте модель в виде изображения, экспортируйте модель в виде SQL-скрипта.

В отчете требуется отобразить:

- схему в виде изображения;
- скопированный запрос, соответствующий созданию этой базы данных:  
вставьте его в какой-либо сервис для хранения фрагментов кода ([pastebin.com](https://pastebin.com), [gist](https://gist.github.com)), сгенерируйте публичную ссылку и вставьте её в отчёт;
- фрагмент запроса, касающийся создания и настройки таблицы invoice.

# Задание 1



<https://pastebin.com/PmsajpD5>

# Задание 1

```
CREATE TABLE IF NOT EXISTS `firstmodel`.`invoice` (  
  `idinvoice` INT NOT NULL AUTO_INCREMENT,  
  `userid` INT NOT NULL,  
  `productid` INT NOT NULL,  
  `price` DECIMAL(10,2) NOT NULL,  
  PRIMARY KEY (`idinvoice`),  
  INDEX `user_idx` (`userid` ASC) VISIBLE,  
  INDEX `prod_idx` (`productid` ASC) VISIBLE,  
  CONSTRAINT `user`  
    FOREIGN KEY (`userid`)  
    REFERENCES `firstmodel`.`user` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `prod`  
    FOREIGN KEY (`productid`)  
    REFERENCES `firstmodel`.`product` (`idproduct`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)
```

# Задание 2

**Задание 2.** Создайте собственную EER-диаграмму и спроектируйте БД с параметрами на основе текста, опубликованного по ссылке:

<https://habr.com/ru/post/175985/>

Будьте внимательны: в тексте не всегда точно описаны параметры столбцов.

Всегда сопоставляйте текст и изображения, которые приводит автор и приводите параметры в соответствие с изображениями таблиц модели.

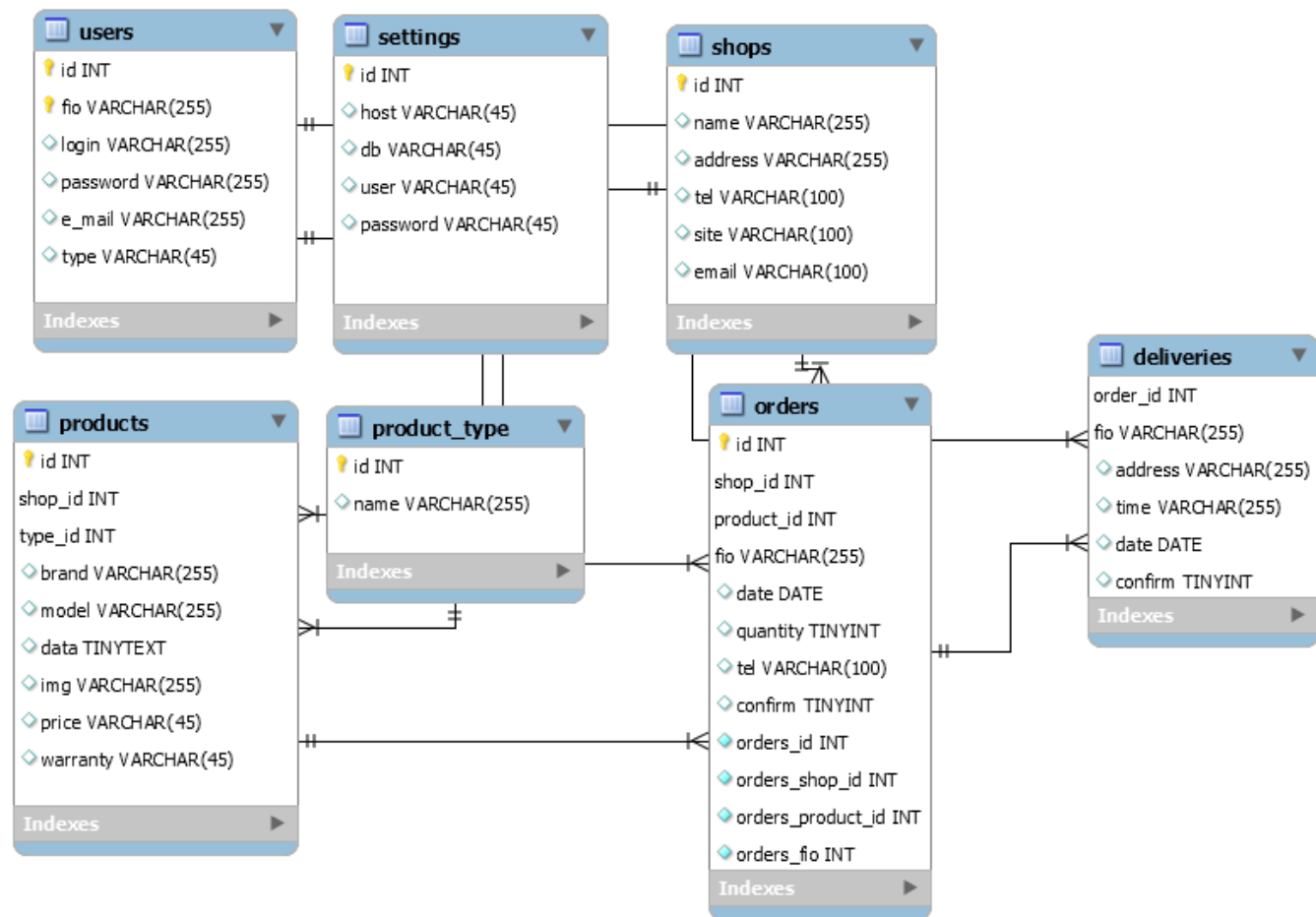
Текст следует читать до абзаца «**Нормальная форма — свойство отношения в реляционной модели данных...**».

Экспортируйте полученную модель в виде изображения, экспортируйте модель в виде SQL-скрипта.

В отчете требуется отобразить:

- схему в виде изображения;
- скопированный запрос, соответствующий созданию этой базы данных:  
вставьте его в какой-либо сервис для хранения фрагментов кода  
([pastebin.com](https://pastebin.com), [gist](https://gist.github.com)), сгенерируйте публичную или секретную ссылку и вставьте её в отчёт;
- фрагмент запроса, касающийся создания и настройки таблицы *Orders*.

# Задание 2



<https://pastebin.com/G6h0hs0f>

# Задание 2

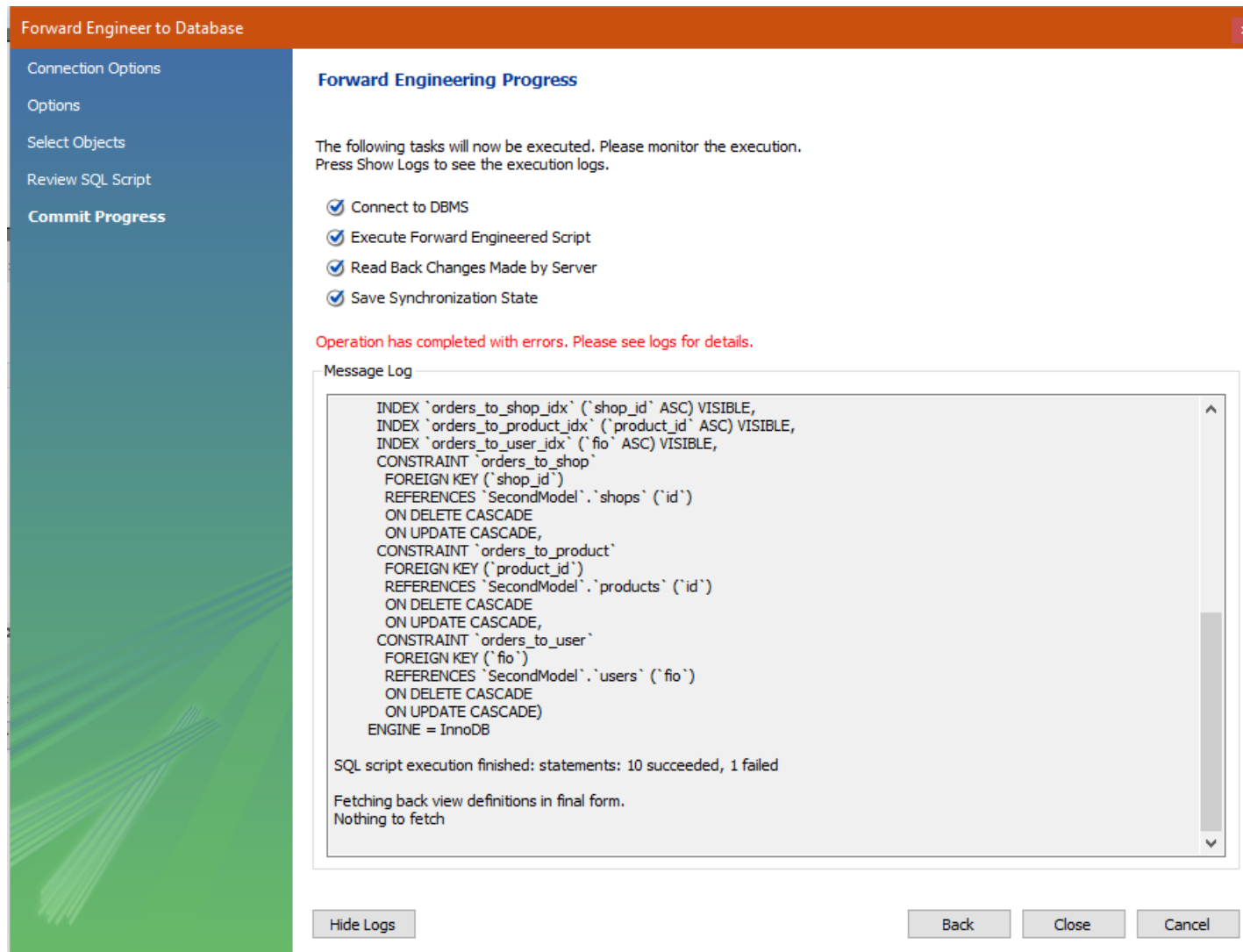
```
CREATE TABLE IF NOT EXISTS `SecondModel`.`orders` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `shop_id` INT NOT NULL,  
  `product_id` INT NOT NULL,  
  `fio` VARCHAR(255) NOT NULL,  
  `date` DATE NULL,  
  `quantity` TINYINT NULL,  
  `tel` VARCHAR(100) NULL,  
  `confirm` TINYINT NULL,  
  `orders_id` INT NOT NULL,  
  `orders_shop_id` INT NOT NULL,  
  `orders_product_id` INT NOT NULL,  
  `orders_fio` INT NOT NULL,  
  PRIMARY KEY (`id`, `shop_id`, `product_id`, `fio`),  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,  
  INDEX `orders_to_shop_idx` (`shop_id` ASC) VISIBLE,  
  INDEX `orders_to_product_idx` (`product_id` ASC) VISIBLE,  
  INDEX `orders_to_user_idx` (`fio` ASC) VISIBLE,  
  CONSTRAINT `orders_to_shop`  
    FOREIGN KEY (`shop_id`)  
    REFERENCES `SecondModel`.`shops` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `orders_to_product`  
    FOREIGN KEY (`product_id`)  
    REFERENCES `SecondModel`.`products` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `orders_to_user`  
    FOREIGN KEY (`fio`)  
    REFERENCES `SecondModel`.`users` (`fio`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)
```

# Задание 3

**Задание 3.** Выполните операцию Database - Forward Engineer и создайте базу данных на вашем сервере. Сделайте скриншот с успешным выполнением этого процесса и вставьте его в отчет.



# Задание 3



# Задание 4

**Задание 4.** Добавьте несколько строк в каждую таблицу созданной базы данных. Попробуйте удалить связанные в нескольких таблицах данные, зафиксируйте, что произошло и опишите текстом (и по возможности дополните скриншотами) в отчёте.

# Задание 4

**CASCADE:** автоматически удаляет или изменяет строки из зависимой таблицы при удалении или изменении связанных строк в главной таблице.

Каскадное удаление позволяет при удалении строки из главной таблицы автоматически удалить все связанные строки из зависимой таблицы. Для этого применяется опция **CASCADE**.

Подобным образом работает и выражение **ON UPDATE CASCADE**. При изменении значения первичного ключа автоматически изменится значение связанного с ним внешнего ключа.