

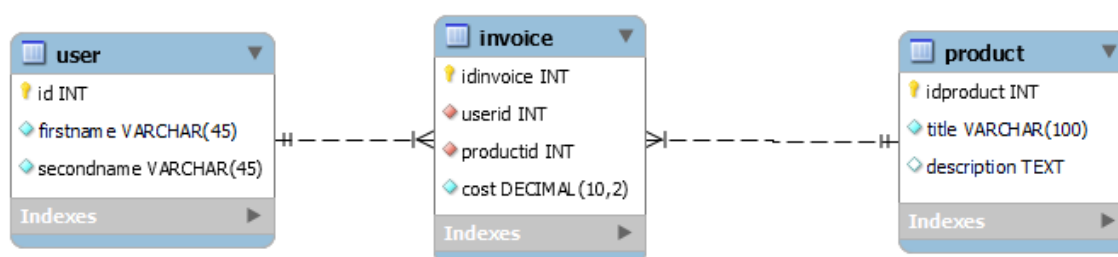
# Лабораторная работа 2. Схема данных. EER-диаграмма

## Задание 1

Повторите действия, демонстрируемые в ролике и создайте диаграмму (модель), созданную во второй половине ролика. Экпортируйте 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, gist), сгенерируйте публичную ссылку и вставьте её в отчёт;  
<https://pastebin.com/50FrWhBh>
- фрагмент запроса, касающийся создания и настройки таблицы invoice.

```
CREATE TABLE IF NOT EXISTS `firstModel`.`invoice` (  
  `idinvoice` INT NOT NULL AUTO_INCREMENT,  
  `userid` INT NOT NULL,  
  `productid` INT NOT NULL,  
  `cost` 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)  
ENGINE = InnoDB
```

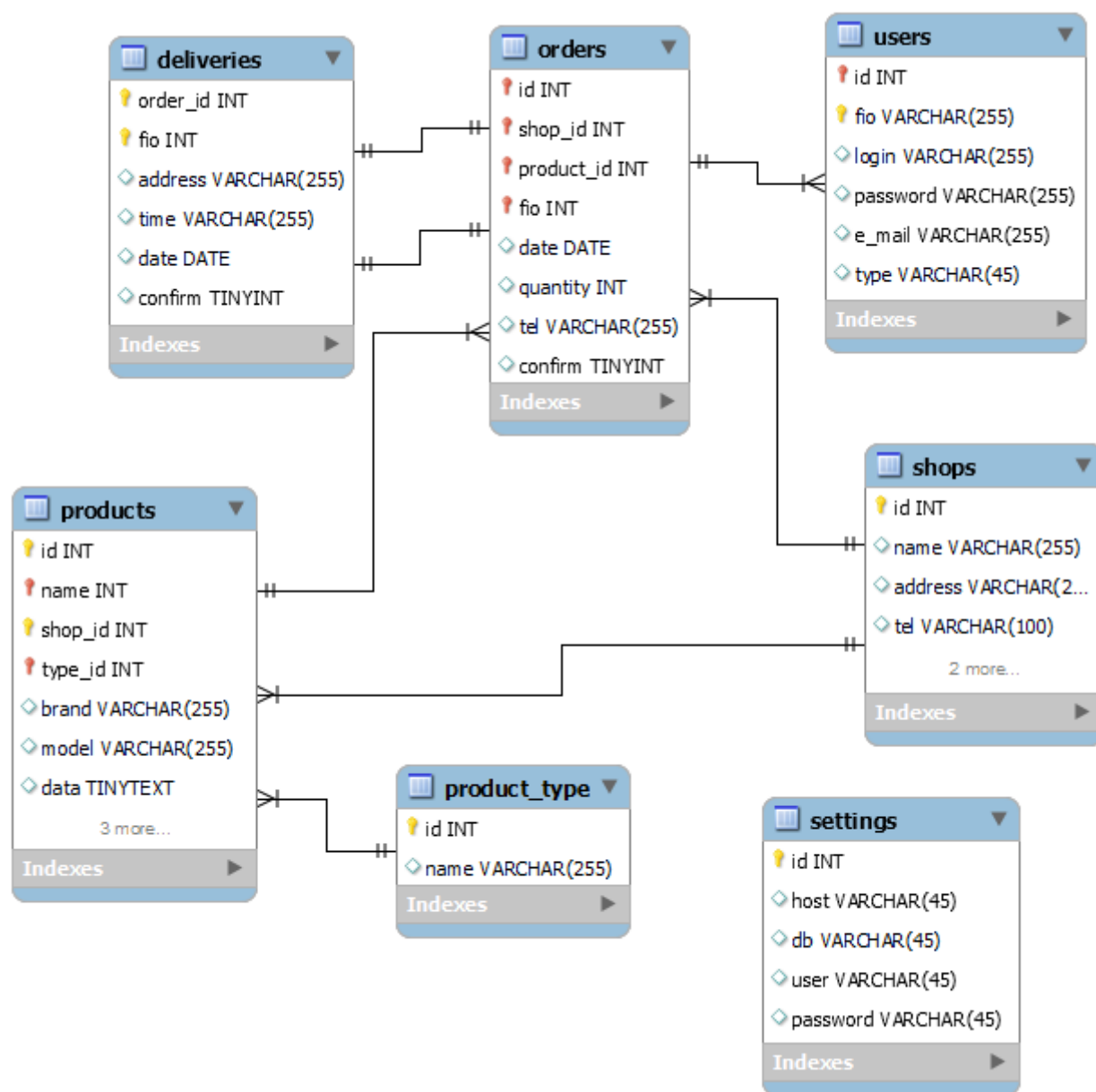
## Задание 2

Создайте собственную EER-диаграмму и спроектируйте БД с параметрами на основе текста, опубликованного по ссылке: <https://habr.com/ru/post/175985/>

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, gist), сгенерируйте публичную или секретную ссылку и вставьте её в отчёт;

<https://pastebin.com/uL6Z7MjS>

- фрагмент запроса, касающийся создания и настройки таблицы Orders.

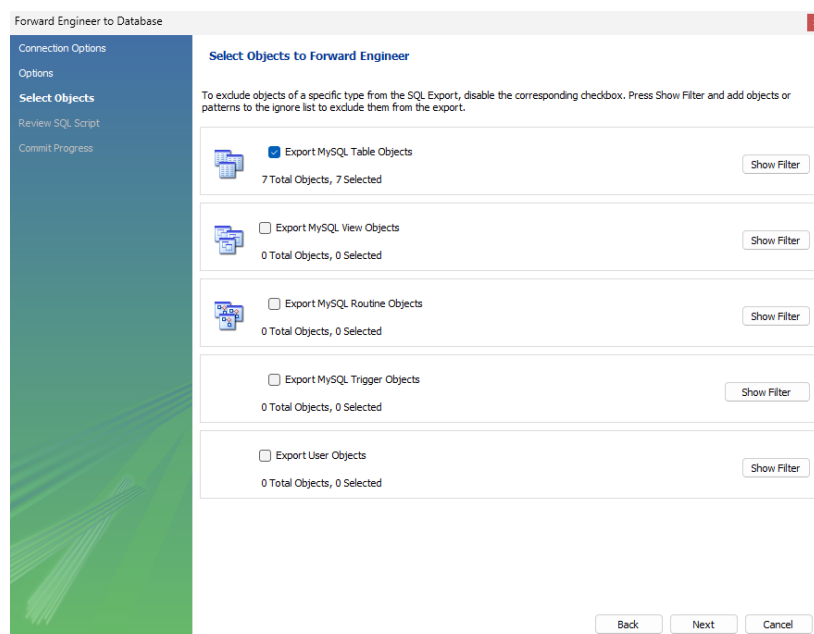
```

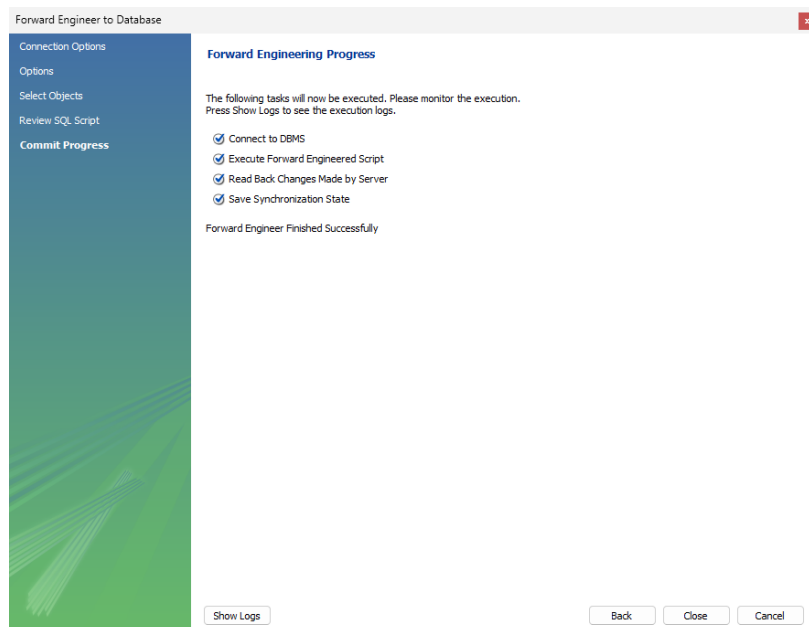
CREATE TABLE IF NOT EXISTS `myChart`.`orders` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `shop_id` INT NOT NULL,
  `product_id` INT NOT NULL,
  `fio` INT NOT NULL,
  `date` DATE NULL,
  `quantity` INT NULL,
  `tel` VARCHAR(255) NULL,
  `confirm` TINYINT NULL,
  PRIMARY KEY (`id`, `shop_id`, `product_id`, `fio`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `orders_to_del(fio)_idx` (`fio` ASC) VISIBLE,
  INDEX `orders_to_del(shop)_idx` (`shop_id` ASC) VISIBLE,
  INDEX `orders_to_prod_idx` (`product_id` ASC) VISIBLE,
  CONSTRAINT `orders_to_del(fio)`
    FOREIGN KEY (`fio`)
      REFERENCES `myChart`.`deliveries` (`fio`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `orders_to_del(shop)`
    FOREIGN KEY (`shop_id`)
      REFERENCES `myChart`.`deliveries` (`order_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `orders_to_shops(id)`
    FOREIGN KEY (`id`)
      REFERENCES `myChart`.`shops` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `orders_to_prod`
    FOREIGN KEY (`product_id`)
      REFERENCES `myChart`.`products` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### 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.



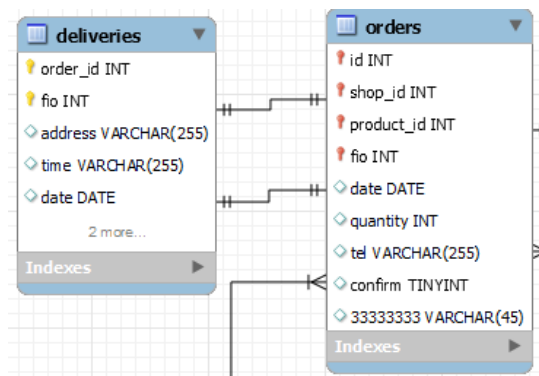


## Задание 4

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.

Удалив связанные данные из двух столбцов, ничего не увидел кроме того, что связь исчезла.

Было:



Стало:

