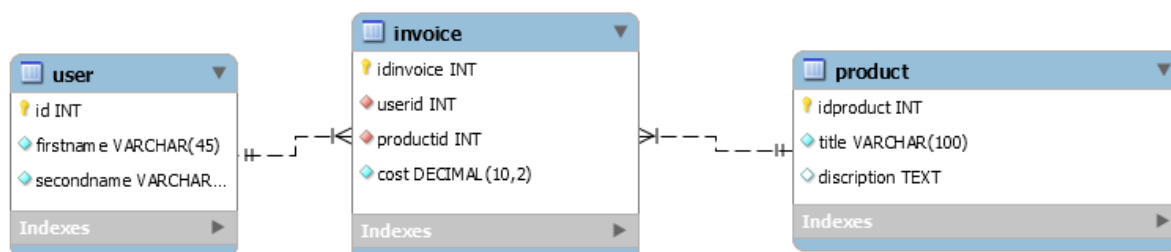


Задание 1

Скопированный запрос:

<https://gist.github.com/Igor-voy/e2de385ffcb3ba2cc3d093778dcace4f>

Схема в виде изображения:



Фрагмент запроса по созданию и настройке таблицы «invoice»:

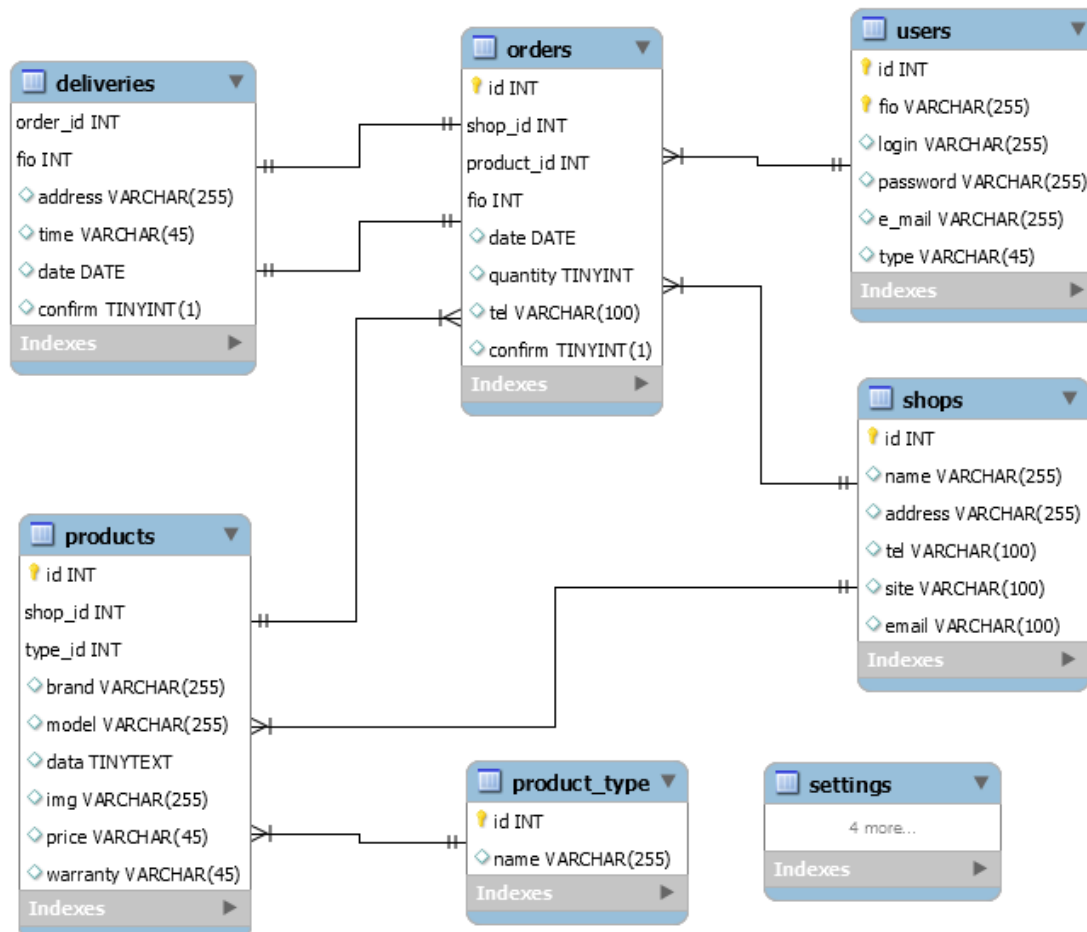
```
CREATE TABLE IF NOT EXISTS `firstModule`.`invoice` (  
  `idinvoice` INT NOT NULL AUTO_INCREMENT,  
  `userid` INT NOT NULL,  
  `productid` INT NOT NULL,  
  `cost` DECIMAL(10,2) NOT NULL,  
  PRIMARY KEY (`idinvoice`),  
  INDEX `user_idx` (`userid` ASC) VISIBLE,  
  INDEX `product_idx` (`productid` ASC) VISIBLE,  
  CONSTRAINT `user`  
    FOREIGN KEY (`userid`)  
      REFERENCES `firstModule`.`user` (`id`)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE,  
  CONSTRAINT `product`  
    FOREIGN KEY (`productid`)  
      REFERENCES `firstModule`.`product` (`idproduct`)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE)  
ENGINE = InnoDB
```

Задание 2

Скопированный запрос:

<https://gist.github.com/Igor-voy/f67543501a43a170eecf2822fec3bb22>

Схема в виде изображения:



Фрагмент запроса по созданию и настройке таблицы «orders»:

```

CREATE TABLE IF NOT EXISTS `habr`.`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` TINYINT NULL,
  `tel` VARCHAR(100) NULL,
  `confirm` TINYINT(1) NULL,
  PRIMARY KEY (`id`, `shop_id`, `product_id`, `fio`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `shop_to_shop_idx` (`product_id` ASC) VISIBLE,
  INDEX `shop_to_shop_idx1` (`shop_id` ASC) VISIBLE,
  INDEX `id_to_fio_idx` (`fio` ASC) VISIBLE,
  CONSTRAINT `id_to_fio`
    FOREIGN KEY (`fio`)
      REFERENCES `habr`.`users` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
  CONSTRAINT `product_to_product`
    FOREIGN KEY (`product_id`)
      REFERENCES `habr`.`products` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
  CONSTRAINT `shop_to_shop`
    FOREIGN KEY (`shop_id`)
      REFERENCES `habr`.`shops` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE)
ENGINE = InnoDB

```

Задание 3

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Forward Engineering Progress

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- ☒ Connect to DBMS
- ☒ Execute Forward Engineered Script
- ☒ Read Back Changes Made by Server
- ☒ Save Synchronization State

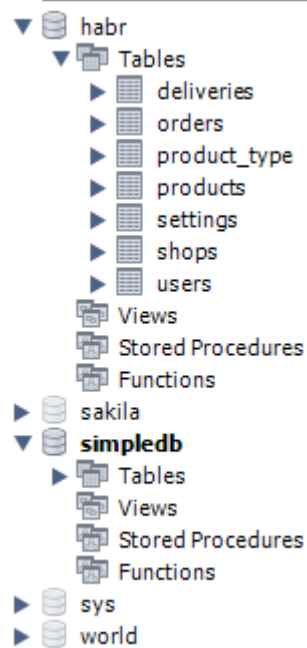
Forward Engineer Finished Successfully

Show Logs

Back

Close

Cancel



Задание 4

Так как таблицы связаны для добавления и удаления данных с помощью «CASCADE», то при добавлении или удалении данных, они будут появляться или исчезать в ключевых полях.

Вывод: в ходе лабораторной работы были получены знания по построению схем данных (EER-диаграмм).