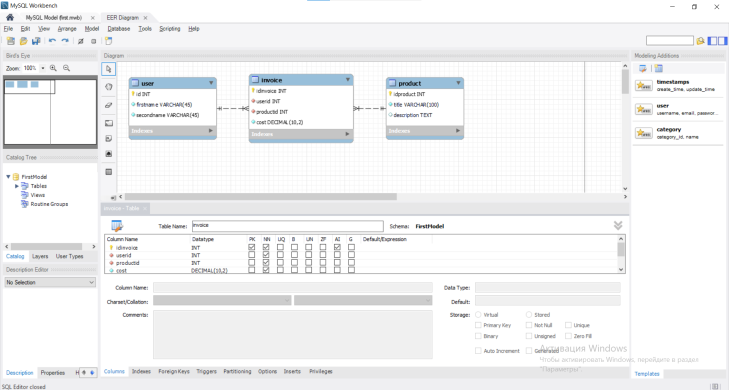
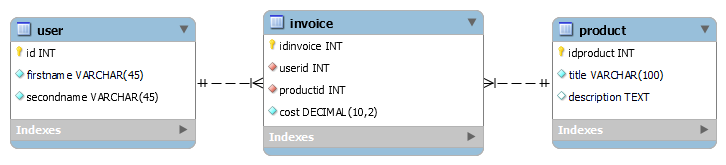
**Лабораторная работа №2**

*Задание 1*

Схема EER диаграммы:





[Ссылка на GIST](https://gist.github.com/XtulenchikX/a600f48f55158a4a910f8098ed488a05)

Фрагмент запроса для создания и настройки таблицы invoice:

-- -----------------------------------------------------

-- Table `FirstModel`.`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`)

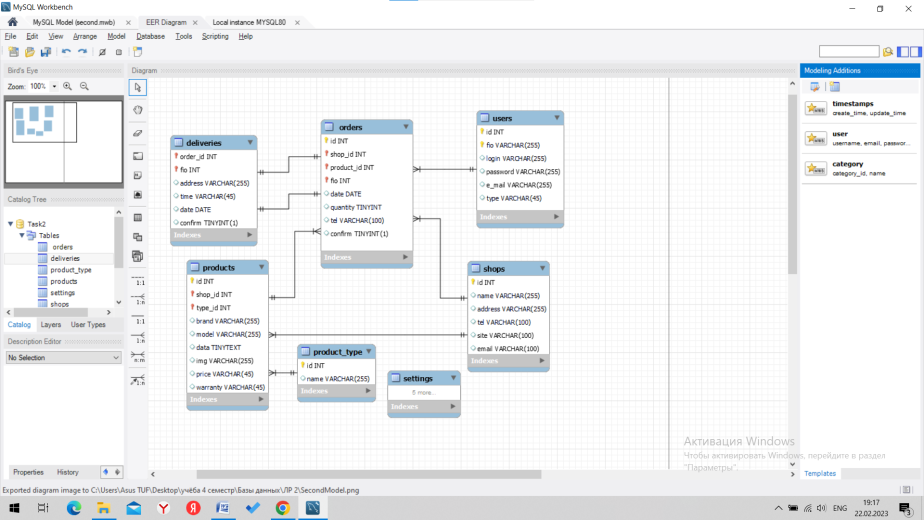
REFERENCES `FirstModel`.`product` (`idproduct`)

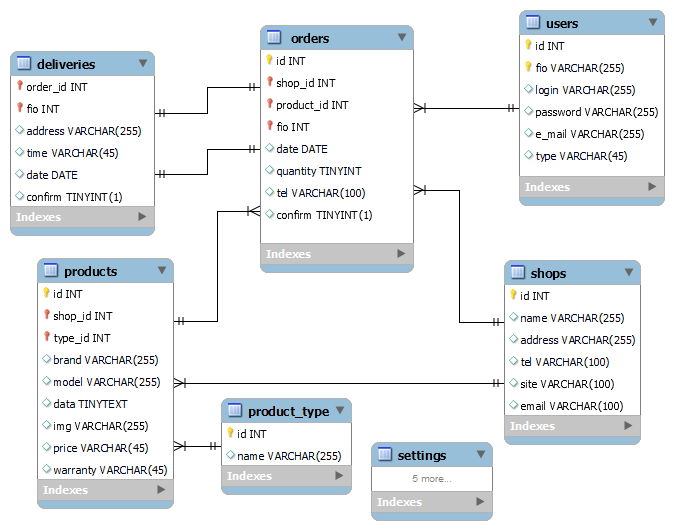
ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

*Задание 2*





[Ссылка на GIST](https://gist.github.com/XtulenchikX/62adb26a3c4f92daf0933237ff6cb76d)

Фрагмент запроса для создания и настройки таблицы Orders:

-- -----------------------------------------------------

-- Table `Task2`.` orders`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Task2`.` 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 `orders\_to\_products\_idx` (`product\_id` ASC) VISIBLE,

INDEX `orders\_to\_shops\_idx` (`shop\_id` ASC) VISIBLE,

INDEX `orders\_to\_users\_idx` (`fio` ASC) VISIBLE,

CONSTRAINT `orders\_to\_products`

FOREIGN KEY (`product\_id`)

REFERENCES `Task2`.`products` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `orders\_to\_shops`

FOREIGN KEY (`shop\_id`)

REFERENCES `Task2`.`shops` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `orders\_to\_users`

FOREIGN KEY (`fio`)

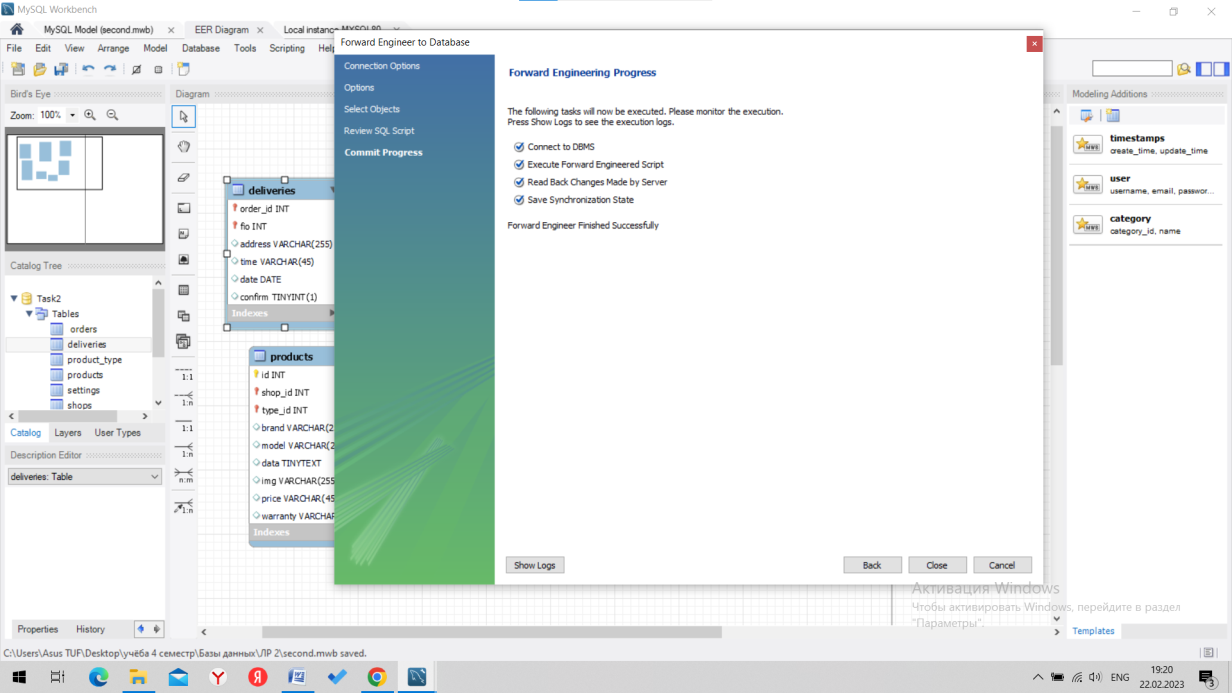
REFERENCES `Task2`.`users` (`id`)

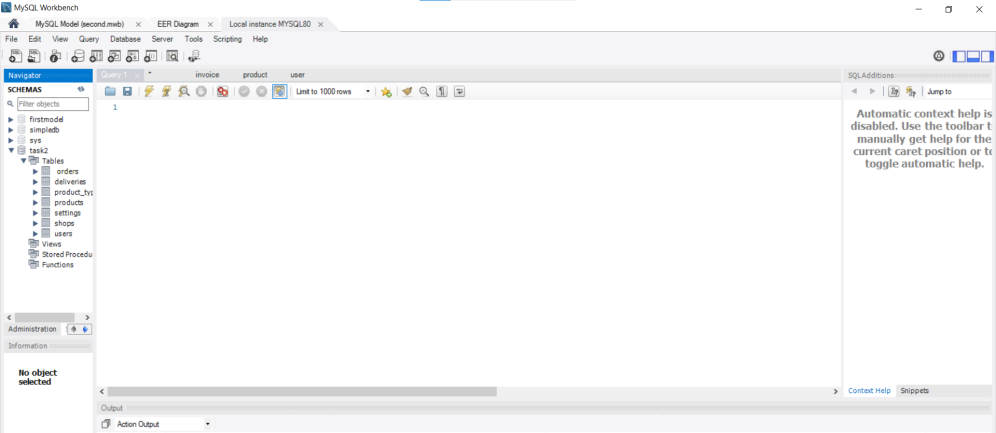
ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

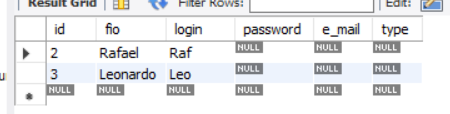
*Задание 3*

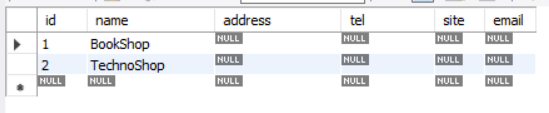


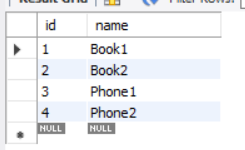


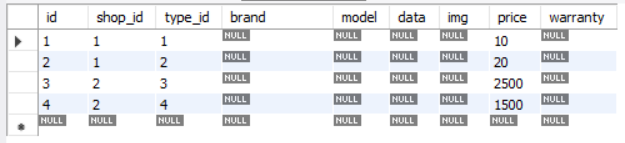
*Задание 4*

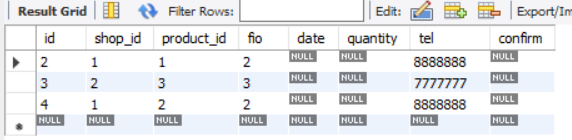
Добавленные в таблицы строки:

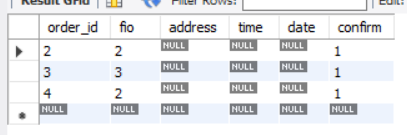








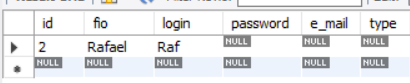




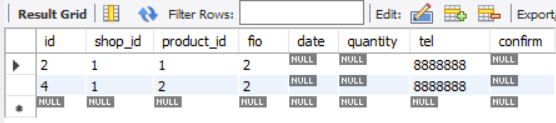
Если мы удаляем строку, первичный ключ которой является внешним ключом для строки в другой таблице нашей базы данных, то будут удалены обе строки.

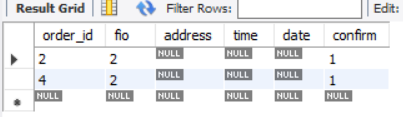
Например:

- Если мы удалим пользователя из таблицы “users”, то все строки связанные с ним в таблицах “orders” и “deliveries” также удалятся (удалили пользователя Leonardo).

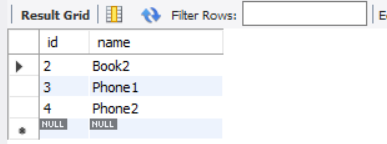


Видим, что заказы и доставки, которые были связаны с id удалённого пользователя, также были удалены.

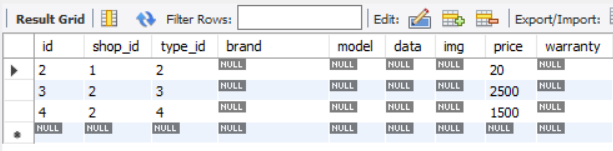


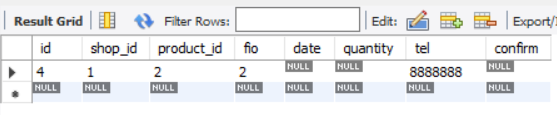


- Если мы удалим строку из таблицы “products\_type”, то все строки связанные с ней в таблицах “products” и “orders” также удалятся (удалили Book1).



Видим, что связанные строки в таблицах “products” и “orders” также удалились.





А также удалилась строка в таблице “deliveries”, которая была связана с удалившейся строкой в таблице “orders”.

