**Практическая работа 6. Создание базы данных**

**Задание:**

К проекту ИС для торговой компании (созданному в предыдущем задании) добавить следующее. Для того, чтобы хранить данные о товарах, поставщиках и ценах, требуется создать базу данных. Создать следующие таблицы (модели в проекте django):

* Товар (столбцы название, описание – текстовый формат данных)
* Поставщик (название компании - текст)
* Склад (товар, поставщик, дата поставки, цена поставки, количество товара на складе на данный момент)

В таблице Склад поля товар и поставщик являются внешними ключами (Foreign key) к соответствующим таблицам

**Пример:**

1. В соответствие с заданием перейдем в файл **models.py** и создадим там следующее описание классов.

from django.db import models  
  
  
class Product(models.Model):  
 name = models.CharField(max\_length=50, verbose\_name=**"Название продукта"**)  
 description = models.CharField(max\_length=100, verbose\_name=**"Описание продукта"**)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
  
class Provider(models.Model):  
 name = models.CharField(max\_length=50, verbose\_name=**"Название компании"**)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
  
class Store(models.Model):  
 product = models.ForeignKey(Product, on\_delete=models.CASCADE, verbose\_name=**"Продукт"**)  
 provider = models.ForeignKey(Provider, on\_delete=models.CASCADE, verbose\_name=**"Компания поставщик"**)  
 date = models.DateField(verbose\_name=**"Дата поставки"**)  
 price = models.IntegerField(verbose\_name=**"Цена"**)  
 quantity = models.IntegerField(verbose\_name=**"Количество в наличии"**)  
  
 def \_\_str\_\_(self):  
 return str(self.product) + str(self.price)

2. Создадим миграции (в терминале напишем команду python manage.py makemigrations)

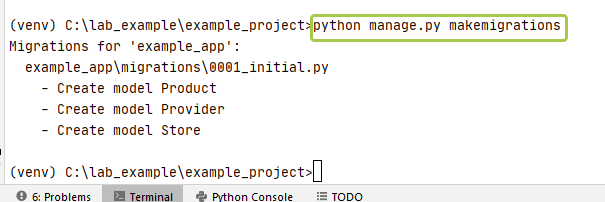


Рисунок. Пример

3. Выполним миграции (в терминале напишем команду python manage.py migrate)

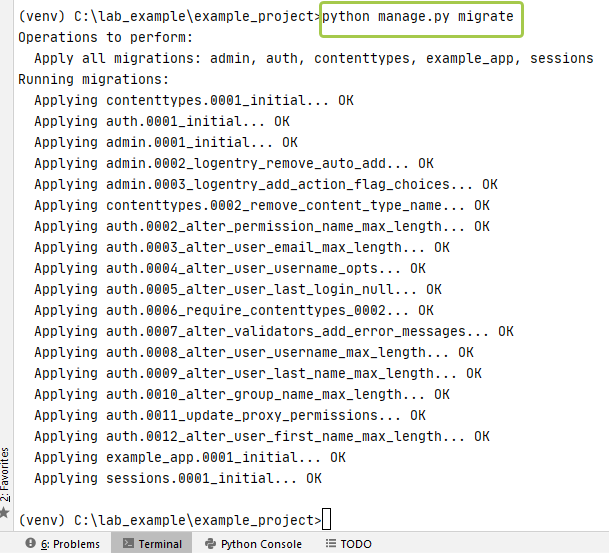


Рисунок. Пример

Таким образом, в результате будут созданы описанные таблицы в базе данных.