

Python задачи

1. Написать код (с помощью регулярных выражений и без них) для удаления из строки незакрытых скобок вместе с их содержимым, если после них нет закрытых блоков: 'esdfd((esdf)(esdf' -> 'esdfd((esdf)'. Написать тесты для этого кода.

2. Запросы Django Даны модели:

```
class Category(models.Model):
    name = models.CharField('Группа товара', max_length=64) ...

class Product(models.Model):
    category = models.ForeignKey(Category, verbose_name='Группа')
    name = models.CharField('Название товара', max_length=128)
    price = models.DecimalField('Стоимость единицы, руб.',
                                max_digits=10,
                                decimal_places=2)
```

а) С помощью Django ORM выбрать товары, цена которых больше или равна 100 руб., сгруппировать по категориям и посчитать количество товаров в каждой категории.

б) То же самое, но оставить лишь категории, в которых строго больше 10 товаров

в) Написать код python, который выводит в консоль перечень всех товаров. Каждая строка должна содержать следующие данные:

- название категории товара,
 - наименование товара, • цена.
- По возможности, минимизировать количество обращений к базе данных и количество передаваемых данных

Additional questions

1. Suppose we have several independent models which have some fields in common. What type of django model inheritance should be used to avoid typing those common fields definitions in each model?

i. Proxy models

ii. Multi-table inheritance

iii. Common fields inheritance iv. Abstract base classes

2. Suppose we have model with a custom manager:

```
class CustomManager(models.Manager):  
    def get_queryset(self):  
        return CustomQuerySet(self.model, using=self._db)
```

```
class Item(models.Model):  
    name = models.CharField('Item', max_length=100)  
    active = models.BooleanField('Active', default=True)  
    objects = CustomManager()
```

Now we want the following code to work:

Item.objects.filter(pk=1).delete() - should set "active" attribute to False on matched records. Item.objects.filter(pk=1).delete_real() - should delete matched records from db.

What is the correct CustomQuerySet implementation?

i.

```
class CustomQuerySet(QuerySet):
```

```
    def delete(self): self.update(active=False)
```

```
    def delete_real(self): super(CustomQuerySet, self).delete_real()
```

ii.

```
class CustomQuerySet(QuerySet):
```

```
    def delete(self): self.update(active=False)
```

```
    def delete_real(self): super(CustomQuerySet, self).delete()
```

iii.

```
class CustomQuerySet(QuerySet):
```

```
def delete(self): self.active = False
```

```
def delete_real(self):
```

```
super(CustomQuerySet, self).delete()
```

iv.

```
class CustomQuerySet(QuerySet):
```

```
def delete(self): self.active = False
```

```
def delete_real(self): super(CustomQuerySet, self).delete_real()
```

3. Suppose we have model:

```
class Person(models.Model):
```

```
name = models.CharField('Item', max_length=100) birthday =  
models.DateField(...)
```

We want to define a model field “birthday” such that django admin interface doesn't allow this field to be empty, but we can create persons with empty birthday using orm (Person.objects.create(name='Name 1') should work).

What is the correct field definition?

i. birthday = models.DateField(null=True, blank=True)

ii. birthday = models.DateField(null=False, blank=True)

iii. birthday = models.DateField(null=True, blank=False)

iv. birthday = models.DateField(null=False, blank=False)

Проектирование

Предложить структуру хранения данных и технологию их обработки для проекта интернет-магазина (упрощённо). В магазине тысячи товаров различного предназначения, брендов и т.д.

В интернет-магазине необходимо предусмотреть систему скидок. Виды скидок:

- Скидка на единичный товар

- Скидка на бренд
- Скидка на группу товаров (группы: предназначение(техника, одежда,...), пол)
- Скидка клиента
Скидка имеет начало и конец действия. У каждого товара в определённый момент может быть ни одной/одна/несколько скидок. Скидка в данный момент рассчитывается как максимум из действующих скидок для товара. Скидку клиента учитывать в процессе покупки товара, при отображении каталога товаров цены с учётом скидки клиента не пересчитывать. Остальные скидки при отображении каталога должны быть показаны. Необходимо, чтобы была возможность сортировки товаров по актуальной цене - с учётом скидок.
- Какие django-приложения для решения задачи предлагаете использовать (готовые движки магазинов не предлагать)?

SQL (postgresql)

Есть таблица phones с полями: phone - varchar
users - int[]

Есть вторая таблица items
id serial
user_id int
status smallint (3 - не продан, 7 - продан, 5 - резерв)

1. Надо написать запрос который на заданные телефоны возвращает количество проданных вещей.
2. Который возвращает в одном запросе количество и проданных, и непроданных.

GIT

Показать команды

- 1) создание и переход на новую ветку `feature_1`
- 2) сделать два коммита
- 3) удалить второй коммит вместе с изменениями
- 4) смержить эту ветку в `master`