```
1. Django Migrations Advanced:
    1.1. Basic Overview:
        - makemigrations
        - migrate
    2.1. Migration Files
        initial = True                  --> only for the fist migration
        dependencies = []               --> depends on the consecutiveness of our migraions in all apps
        operations                      --> things to be applied as migration
    2.2. Applying Migrations
        - from all apps:
            python manage.py migrate
        - from one app:
            python manage.py migrate main_app
        - specific migration:
            python manage.py migrate main_app 0001
    2.3. Reversing migration:
        - reverse certain migration:
            python manage.py migrate main_app 0001 (app name and the number of migration  you need to revert to)
        - reverse all migrations
            python manage.py migrate main_app zero
    2.4. Showing migrations:
        - python manage.py showmigrations
        - python manage.py showmigrations main_app
    2.5. Optimizing Migrations Number and Size:
        - python manage.py squashmigrations main_app 0238
    2.6. Print SQL for the migration:
        - python manage.py sqlmigrate main_app 0002
    2.7. Custom/Data Migrations
        - python manage.py makemigrations --empty main_app

        def add_full_name(apps, schema_editor):
            Employee = apps.get_model("main_app", "Employee")
            for employee in Employee.objects.all():
                employee.full_name = f"{employee.first_name} {employee.last_name}"
                employee.save()

        from django.db import migrations
        class Migration(migrations.Migration):
            dependencies = [
                "main_app", "0002_employee_full_name"),
            ]
            operations = [
                migrations.RunPython(add_full_name)
            ]

            ###Example for data migration###
            ! Note: never import the model on the first line but do it as shown below in the function:

            class Product(models.Model):
                # add barcode to existing class and migrate
                barcode = models.IntegerField(unique=True)


            def add_barcode(app, schema_editor):
```

```
                        Product = app.get_model("main_app", "Product")
                        all_products = Product.objects.all()
                        all_barcodes = random.sample(range(1, 1000),
len(all_products))
                        for i in range(len(all_products)):
                              product = all_products[i]
                              product.barcode = all_barcodes[i]
                              product.save()


                  def reverse_add_barcode(app, schema_editor):
                        Product = app.get_model("main_app", "Product")
                         # Product.objects.all.update(barcode=0) --> Option 1, or
as written below#
                        for product in Product.objects.all():
                              product.barcode = 0
                              product.save()

                  class Migration(migrations.Migration):
                        dependencies = [
                              ('main_app', '0003_alter...'),
                        ]
                        operations = [
                              migrations.RunPython(add_barcode,
reverse_code=reverse_add_barcode),
                        ]
```

2. Django Admin Site:
    - 2.1. Introduction
    - 2.2. Superuser
    - 2.3. Customization

```
      @admin.register(Employee)
      class EmployeeAdmin(admin.ModelAdmin):
            pass

            #display the model fields
            list_display = ['job_title', 'first_name', 'last_name']
            #add filters to models
            list_filter = ['job_level]
            #add search box with filed names that will be searched
            search_fields = ['email_address']
            #make layout changes
            fields = [('first_name'), ('last_name'), 'email_address']
            fieldsets = (
                  ('Personal Info',
                        {'fields': (...)}),
                  ('Advanced Options',
                        {'classes: ('collapse',),
                        {'fields': (...)}),
            )
            date_hierarchy = 'created_on'
```