

1. CRUD Operations:
 - Introduciton
 - Importance
2. QuerySet:
 - lazy evaluation
 - retrieving objects
 - chaining filters
 - query related objects(fk)
 - aggreagation and annotation
 - ordering
 - pagination
3. Django Simple Queries:
 - Default Object Manager: 'objects'
 - Create, Select, Update and Delete
 - Updating a record from the DB table:

```
def update_first_employee():  
    first_employee = Employee.objects.first()  
    first_employee.job_level = 'Sr.'  
    first_employee.save()
```
 - Common Object Manager Methods:
 - all()
 - first()
 - last()
 - get(**kwargs)
 - crete(**kwargs)
 - filter(**kwargs)
 - exclude(**kwargs)
 - order_by(*fields)
 - create()
 - save()
4. Django Shell and SQL Logging

```
#1.  
from main_app.models import Student  
  
def add_students():  
    Student.objects.create(  
        student_id='FC5204',  
        first_name='John',  
        last_name='Doe',  
        birth_date='1995/05/15',  
        email='john.doe@university.com',  
    )  
  
    student2 = Student(  
        student_id='FE0054',  
        first_name='Jane',  
        last_name='Smith',  
        email='jane.smith@university.com',  
    )  
    student2.save()  
  
    Student.objects.create(  
        student_id='FH2014',  
        first_name='Alice',
```

```
        last_name='Johnson',
        birth_date='1998/02/10',
        email='alice.johnson@university.com',
    )
```

```
Student.objects.create(
    student_id='FH2015',
    first_name='Bob',
    last_name='Wilson',
    birth_date='1996/11/25',
    email='bob.wilson@university.com',
)
```

#2.

```
def get_student_info():
    students_data = []
    for student in Student.objects.all():
        students_data.append(
            f"Student N:{student.student_id}:"
            f"{student.first_name} "
            f"{student.last_name};"
            f"Email: {student.email}"
        )
    return "\n".join(students_data)
print(get_student_info())
```

#3.

```
def update_students_emails():
    for student in Student.objects.all():
        new_email = student.email.replace('university.com', 'uni-students.com')
        student.email = new_email
        student.save()
```

#4.

```
def truncate_students():
    Student.objects.all().delete()
```