Database access using Django. Questions.

1. **Indicate an example of database where it is better to use a relational database better than a NoSQL database and explain why.**

   Un ejemplo claro sería la manipulación de datos de un cliente que tiene un carrito de la compra o pedidos realizados en una pagina web. En ellas tienes la tabla cliente, tabla producto y una tabla de pedidos y carrito. Estas dos ultimas tablas hacen que sea menos filas de información que si fuera no relacional, por lo tanto, hace que ocupe menos espacio en la bbdd, es mas fácil de manipular, buscar, y ampliar las filas.

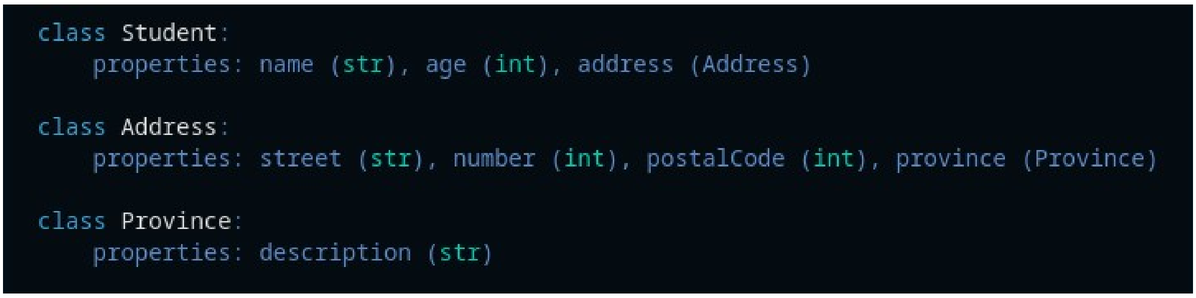
2. **If we have the following tables in a relational database:**

   student: idStudent, idAddress (foreign key to table address), name, age.

   address: idAddress, street, number, postalCode, idProvince (foreign key to table province)

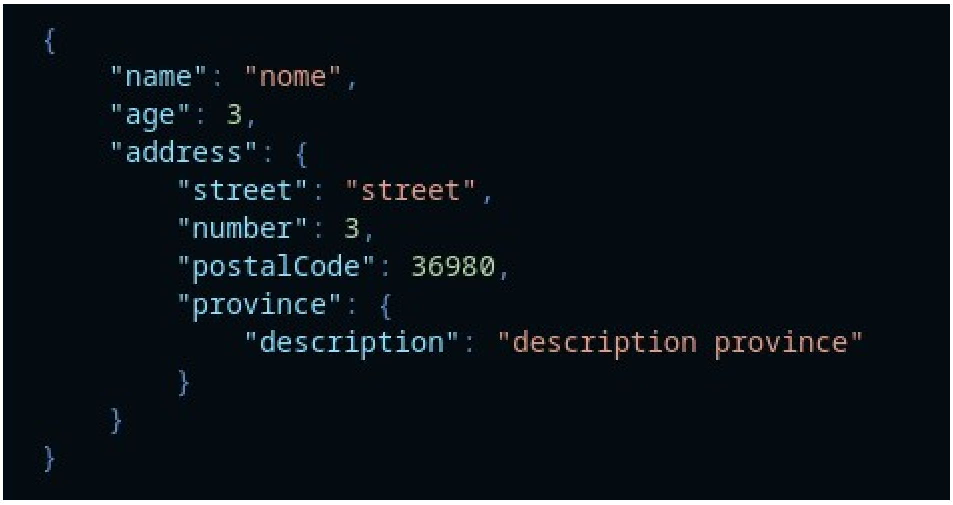
   province: idProvince, description.

   - **Indicate the equivalence in an Object Oriented Model.**



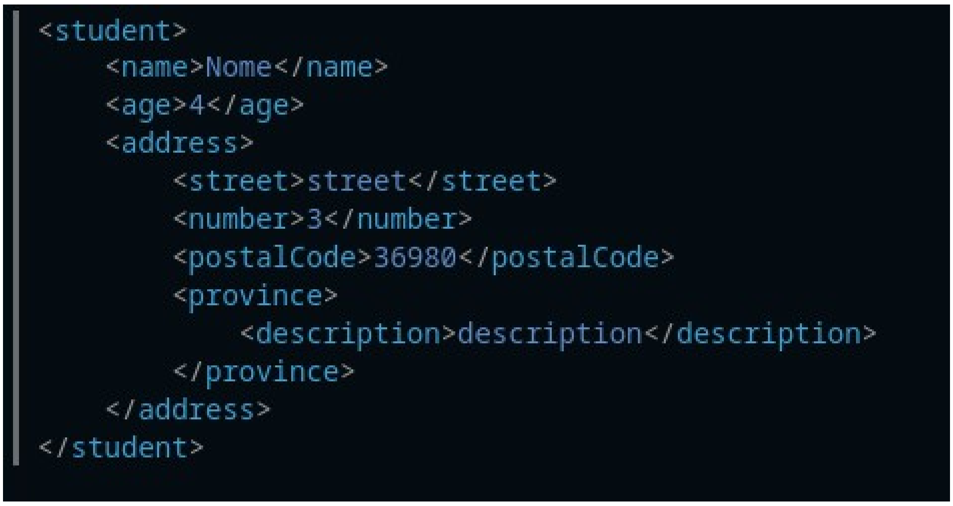
```
class Student:
    properties: name (str), age (int), address (Address)

class Address:
    properties: street (str), number (int), postalCode (int), province (Province)

class Province:
    properties: description (str)
```

   - **Indicate the equivalence in a documental database that uses JSON files.**



```
{
    "name": "nome",
    "age": 3,
    "address": {
        "street": "street",
        "number": 3,
        "postalCode": 36980,
        "province": {
            "description": "description province"
        }
    }
}
```

- **Indicate the equivalence using a xml file.**



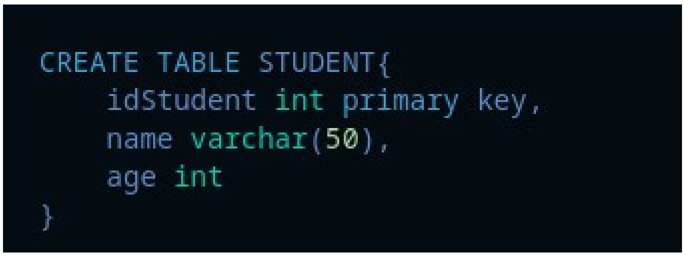
```
<student>
    <name>Nome</name>
    <age>4</age>
    <address>
        <street>street</street>
        <number>3</number>
        <postalCode>36980</postalCode>
        <province>
            <description>description</description>
        </province>
    </address>
</student>
```

3. **Indicate three advantages of using a database better than a file to store information.**

- Permiten una lectura mas fácil.
- Son mas seguros al tener un sistema de autentificación.
- La manipulación de los datos se realiza de manera mas sencilla

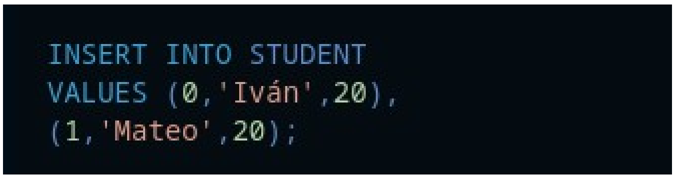
4. **Indicate the SQL code to perform the following actions:**

- **Create a table with name student and fields: idStudent (integer, PK), name (string), age (number)**



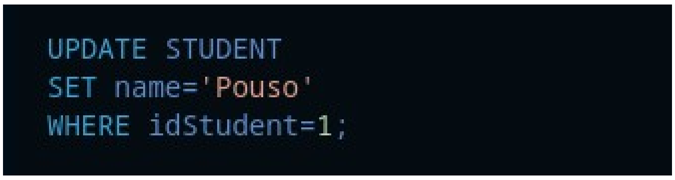
```
CREATE TABLE STUDENT{
    idStudent int primary key,
    name varchar(50),
    age int
}
```

- **Add two rows of data in the previous table.**



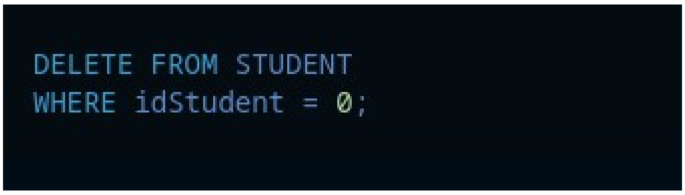
```
INSERT INTO STUDENT
VALUES (0,'Iván',20),
(1,'Mateo',20);
```

- **Modify the second row of data that you added.**



```
UPDATE STUDENT
SET name='Pouso'
WHERE idStudent=1;
```

- **Delete the first row of data in that table.**



```
DELETE FROM STUDENT
WHERE idStudent = 0;
```

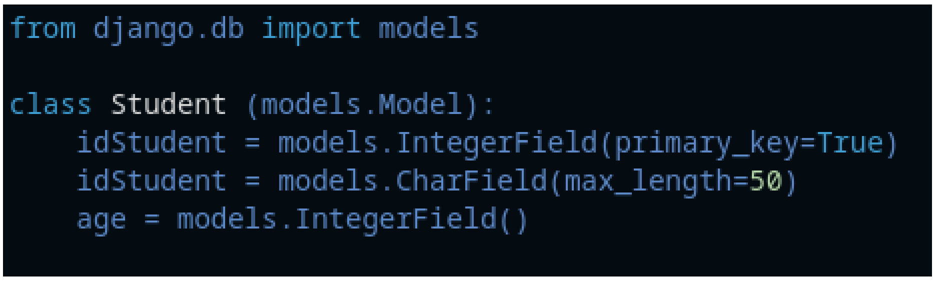
5. **What is a Django Model?**

   Un django model es una tabla en la base de datos generada automaticamente por django. Estas tablas se generan a partir de las clases que se crean en el fichero models.py de django

6. **What does “object-relational mapping” mean?**

   Es el proceso que hay entre la transformación de una fila de datos de una tabla a un objeto entendible para el lenguaje de programación pertinente.

7. **Define the Django model that is equivalent to the table in the exercise 4. Indicate which actions must be performed so that this model is mapped to a relational database.**

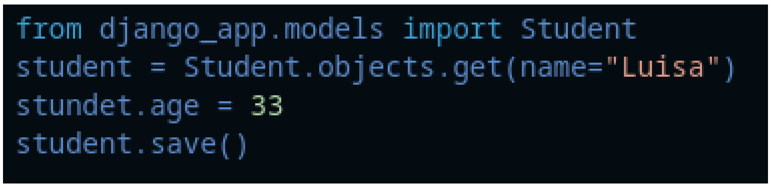


```
from django.db import models

class Student (models.Model):
    idStudent = models.IntegerField(primary_key=True)
    idStudent = models.CharField(max_length=50)
    age = models.IntegerField()
```

8. **What is the purpose of the field validators that we can specify in the Django models?**

   Chequear que los valores entregados con las especificaciones establecidas en el campo del modelo antes de mandarlo a la bbdd.

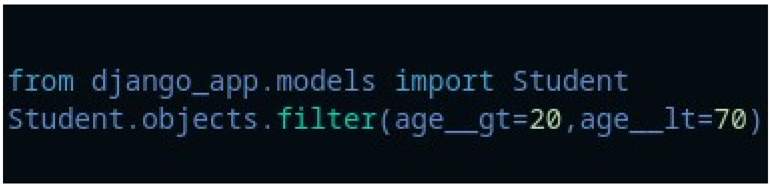
9. **If I have a student with name “Luisa” in the database. Write the Django code that we must execute in order to modify it.**



```
from django_app.models import Student
student = Student.objects.get(name="Luisa")
stundet.age = 33
student.save()
```

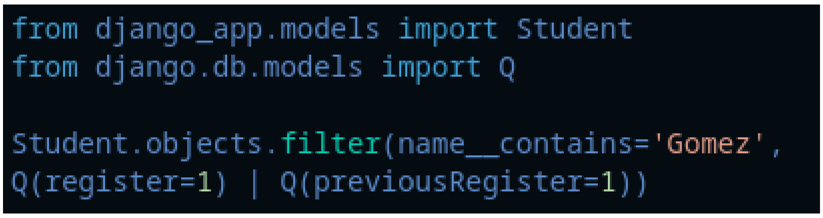
10. Write the code of the filters that are equivalent to the following SQL sentences:

- **select * from student where age between 20 and 70;**



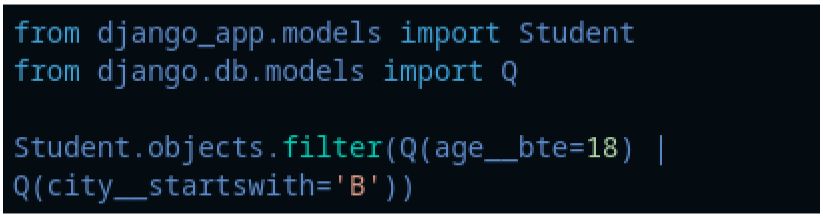
```
from django_app.models import Student
Student.objects.filter(age__gt=20,age__lt=70)
```

- **select * from student where name like '%Gomez%' and (register = 1 or previousRegister = 1);**



```
from django_app.models import Student
from django.db.models import Q

Student.objects.filter(name__contains='Gomez',
Q(register=1) | Q(previousRegister=1))
```

- **select * from customer where age >= 18 or city = 'B%';**



```
from django_app.models import Student
from django.db.models import Q

Student.objects.filter(Q(age__bte=18) |
Q(city__startswith='B'))
```

11. Describe with your own words the following concepts:

- **Persistent Data.**

  Datos que existen incluso después de que el programa ejecutado finalice.

- **Temporary Data.**

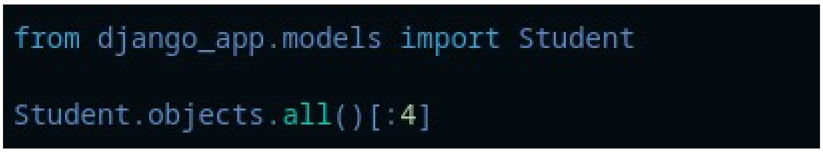
  Datos que solo existen durante durante un corto periodo (memoria RAM), y son eliminados después de finalizar el programa o después de apagar el dispositivo.

- **Semi-Persistent Data.**

  Datos almacenados durante un tiempo en especifico (cookies).

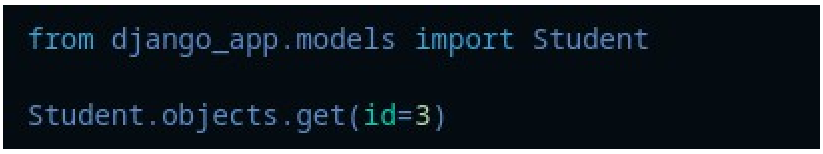
12. Write the django code to perform the following tasks:

- **Get all registers from the table 'student'. Show only the 4 first registers.**



```
from django_app.models import Student

Student.objects.all()[:4]
```

- **Get from the previous table the register with id=3.**



```
from django_app.models import Student

Student.objects.get(id=3)
```

13. We can choose between two ways to operate on the database:

Case 1:

bestsellers = Book.objects.filter(is_bestselling=True)

amazing_bestsellers = bestsellers.filter(rating__gt=4)

print (bestsellers)

print (amazing_bestsellers)

Case 2:

print (Book.objects.filter(is_bestselling=True))

print (Book.objects.filter(is_bestselling=True, rating__gt=4))

- **How many times does Django access the database in “Case 1”? And how many times in “Case 2”?**

  En el primero accede 1 vez a la bbdd y en el segundo caso 2 veces.

- **Indicate the instruction in case 1 when the program access the database.**

  En la primera linea

- **Which case gives a better performance?**

  El primero, por que realiza menos llamadas a la bbdd. La segunda variable busca a partir de los datos ya buscados y almacenados en memoria, por lo tanto, consigue los datos mas rápido.

14. Explain with your own words what is the “Cache” and what is its purpose.

Es un tipo de memoria de acceso más rápido que se usa para encontrar datos que se buscan con frecuencia. Mejora el rendimiento y reduce el tiempo de búsqueda.