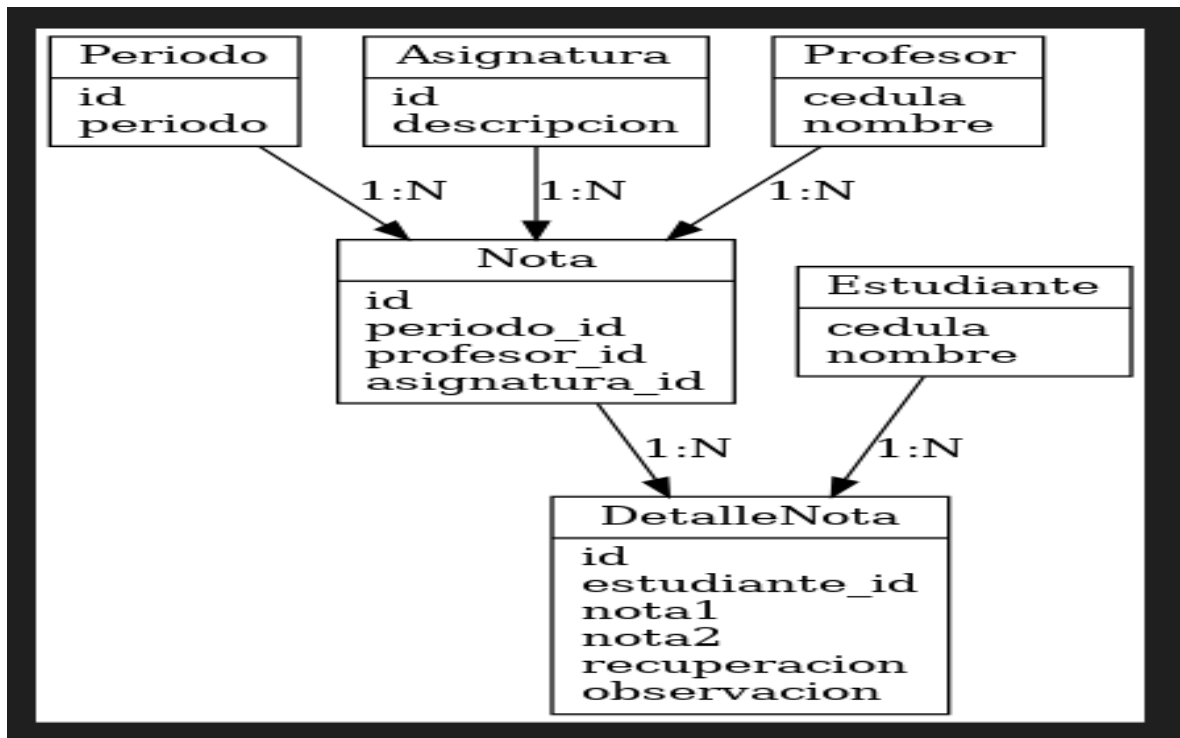


Caso de Estudio: Sistema de Gestión Académica

Escenario

Una universidad está desarrollando un sistema de gestión académica para administrar sus periodos académicos, asignaturas, profesores, estudiantes y las notas de los estudiantes. Como parte de la implementación inicial del sistema, necesitas insertar datos de ejemplo para probar las funcionalidades del sistema. El modelo entidad relación del caso de estudio es el siguiente:



Objetivos

- Insertar datos iniciales para las entidades `Periodo`, `Asignatura`, `Profesor` y `Estudiante` utilizando `bulk_create`.
- Insertar datos iniciales para las entidades `Nota` y `DetalleNota` utilizando los métodos `.create()` y `.save()`.
- Consultar registros de las entidades.

Pasos a Seguir

Insertar registros

Bulk_create

1. **Insertar 10 registros en la tabla `Periodo`**: Crear 10 periodos académicos.
2. **Insertar 10 registros en la tabla `Asignatura`**: Crear 10 asignaturas diferentes.
3. **Insertar 10 registros en la tabla `Profesor`**: Crear 10 profesores con sus identificaciones y nombres.

4. **Insertar 10 registros en la tabla `Estudiante`:** Crear 10 estudiantes con sus identificaciones y nombres.

.Create() y .save()

5. **Insertar 10 registros en la tabla `Nota`:** Registrar 10 notas vinculadas a los periodos, profesores y asignaturas.
6. **Insertar 10 registros en la tabla `DetalleNota`:** Registrar los detalles de las notas para 10 estudiantes de la opción 5

Consultar basicas:

1. Seleccionar todos los estudiantes cuyo nombre comienza con 'Est':
2. Seleccionar todos los profesores cuyo nombre contiene 'or':
3. Seleccionar todas las asignaturas cuya descripción termina en '10':
4. Seleccionar todas las notas con `nota1` mayor que 8.0:
5. Seleccionar todas las notas con `nota2` menor que 9.0:
6. Seleccionar todas las notas con `recuperacion` igual a 9.5:

Consultas usando condiciones lógicas (AND, OR, NOT)

7. Seleccionar todos los estudiantes cuyo nombre comienza con 'Est' y su cedula termina en '1':
8. Seleccionar todas las asignaturas cuya descripción contiene 'Asig' o termina en '5':
9. Seleccionar todos los profesores cuyo nombre no contiene 'or':
10. Seleccionar todas las notas con `nota1` mayor que 7.0 y `nota2` menor que 8.0:
11. Seleccionar todas las notas con `recuperacion` igual a None o `nota2` mayor que 9.0:

Consultas usando funciones numéricas

12. Seleccionar todas las notas con `nota1` entre 7.0 y 9.0:
13. Seleccionar todas las notas con `nota2` fuera del rango 6.0 a 8.0:
14. todas las notas cuya `recuperacion` no sea None:

Consultas usando funciones de fecha (asumiendo que los modelos incluyen campos de fecha)

15. Seleccionar todas las notas creadas en el último año:
16. Seleccionar todas las notas creadas en el último mes:
17. Seleccionar todas las notas creadas en el último día:
18. Seleccionar todas las notas creadas antes del año 2023:
19. Seleccionar todas las notas creadas en marzo de cualquier año:

Consultas combinadas con funciones avanzadas

20. Seleccionar todos los estudiantes cuyo nombre tiene exactamente 10 caracteres:

21. Seleccionar todas las notas con `nota1` y `nota2` mayores a 7.5:
22. Seleccionar todas las notas con `recuperacion` no nula y `nota1` mayor a `nota2`:
23. Seleccionar todas las notas con `nota1` mayor a 8.0 o `nota2` igual a 7.5:
24. Seleccionar todas las notas con `recuperacion` mayor a `nota1` y `nota2`:

Consultas con subconsultas y anotaciones

25. Seleccionar todos los estudiantes con al menos una nota de recuperación:
26. Seleccionar todos los profesores que han dado una asignatura específica:
27. Seleccionar todas las asignaturas que tienen al menos una nota registrada:
28. Seleccionar todas las asignaturas que no tienen notas registradas:
29. Seleccionar todos los estudiantes que no tienen notas de recuperación:
30. Seleccionar todas las notas cuyo promedio de `nota1` y `nota2` es mayor a 8.0:
31. Seleccionar todas las notas con `nota1` menor que 6.0 y `nota2` mayor que 7.0:
32. Seleccionar todas las notas con `nota1` en la lista [7.0, 8.0, 9.0]:
33. Seleccionar todas las notas cuyo `id` está en un rango del 1 al 5:
34. Seleccionar todas las notas cuyo `recuperacion` no está en la lista [8.0, 9.0, 10.0]:
35. Suma de todas las notas de un estudiante:
36. Nota máxima obtenida por un estudiante:
37. Nota mínima obtenida por un estudiante:
38. Contar el número total de notas de un estudiante:
39. Promedio de todas las notas de un estudiante sin incluir recuperación

Consultas con subconsultas con los modelos relacionado. Aplicar relaciones inversas donde sea necesario

40. Dado un estudiante obtener todas sus notas con el detalle de todos sus datos relacionados:
41. Obtener todas las notas de un período específico:
42. Consultar todas las notas de una asignatura dada en un período:
43. Obtener todas las notas de un profesor en particular:
44. Consultar todas las notas de un estudiante con notas superiores a un valor dado:
45. Obtener todas las notas de un estudiante ordenadas por período:
46. Consultar la cantidad total de notas para un estudiante:
47. Calcular el promedio de las notas de un estudiante en un período dado:
48. Consultar todas las notas con una observación específica:
49. Obtener todas las notas de un estudiante ordenadas por asignatura:

Sentencias Update:

- 50. Actualizar nota1 para alumnos con nota1 < 20:
- 51. Actualizar nota2 para alumnos con nota2 < 15:
- 52. Actualizar recuperación para alumnos con recuperación < 10:
- 53. Actualizar observación para alumnos que hayan aprobado:
- 54. Actualizar todas las notas en un período específico:

Sentencias delete

- 55. Eliminar físicamente todas las notas de un estudiante:
- 56. Eliminar lógicamente todas las notas de un estudiante (en el campo `state` que indica si el registro está activo o no):
- 57. Eliminar físicamente todas las notas de un período específico:
- 58. Eliminar lógicamente todas las notas de un período específico:
- 59. Eliminar físicamente todas las notas que tengan una nota1 menor a 10:

Sentencias crud:

- 60. Crea un registro de notas de un estudiante, simulando una inserción de los datos tal como se explicó en el ejercicio de la creación de una factura con su detalle de productos en el archivo `orm.py` de la clase impartida.

Nota: A todos los modelos añadir los atributos: `user`, `created`, `update` y `state`. Además, añada en todos los modelos un manager de `state=true`. También un método `delete()` de borrado lógico(`state=False`) tienen que estar en todos los modelos y otros métodos que usted considere pertinente. Recuerde se valorará la eficiencia de las sentencias del ORM. Para poder realizar esta tarea tiene que crear un proyecto y una app core con los modelos y el archivo `orm.py`. También se revisará la configuración de `tailwindcss` en el proyecto realizado.