

-- Ejercicio 1

```
SELECT m2.Nombre, m2.Especialidad, ifnull(m1.Nombre, 'Non ten supervisor') AS  
NombreSupervisor, m1.Especialidad AS EspecialidadSupervisor  
FROM medicos as m1  
JOIN medicos as m2 ON m1.id = m2.supervisor ;
```

-- Ejercicio 2

```
SELECT p.Nombre, m.Nombre  
FROM pacientes AS p  
JOIN medicos AS m ON p.idMedico = m.id  
ORDER BY p.Nombre ASC, m.Nombre ASC;
```

-- Ejercicio 3

```
SELECT p.Nombre AS NombrePaciente, m.Nombre AS NombreMedico, 'PAC'  
FROM pacientes AS p  
JOIN medicos AS m ON p.idMedico = m.id  
UNION  
SELECT m2.Nombre, m1.Nombre AS NombreSupervisor, 'MEC'  
FROM medicos as m1  
JOIN medicos as m2 ON m1.id = m2.supervisor;
```

-- Ejercicio 4

```
SELECT DISTINCT Tipo, COUNT(Tipo) AS Cantidad  
FROM habitaciones  
GROUP BY Tipo;
```

-- Ejercicio 5

```
SELECT Especialidad, AVG(salario) AS mediaSalarios  
FROM medicos  
GROUP BY Especialidad  
HAVING count(Especialidad) > 2  
ORDER BY AVG(salario) DESC;
```

-- Ejercicio 6

```
SELECT MAX(m.Salario) AS SalrioMaximo, Min(m.Salario) AS SalarioMinimo  
FROM medicos AS m  
WHERE Especialidad LIKE 'Cardiologia' AND YEAR(FechaTitulo) = YEAR(FechaTitulo);
```

-- Ejercicio 7

```
SELECT p.Nombre  
FROM pacientes AS p  
WHERE idHabitacion IN (SELECT id  
                        FROM habitaciones  
                        WHERE Tipo LIKE 'Individual');
```

-- Ejercicio 8

```
SELECT Nombre, Especialidad, date_format(FechaTitulo, '%D%M%Y')  
FROM medicos  
WHERE salario > (SELECT AVG(salario)  
                 FROM medicos)  
ORDER BY Especialidad, FechaTitulo;
```

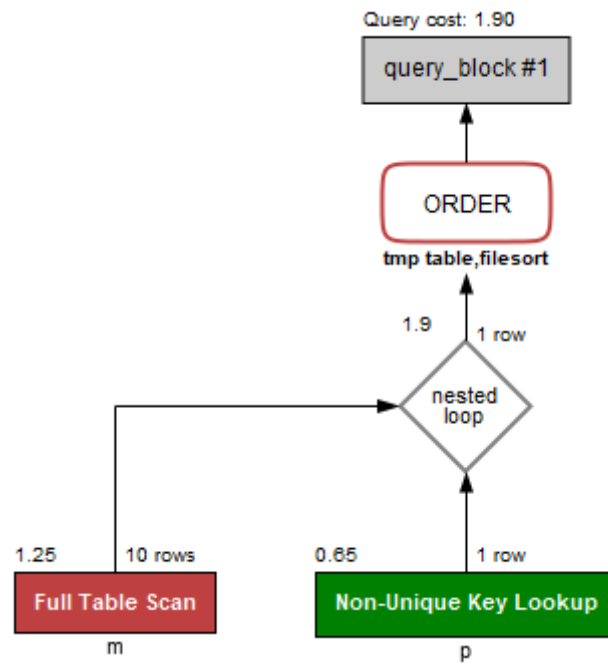
```
-- Ejercicio 9
SELECT Nombre, Especialidad, Salario
FROM medicos
WHERE salario > (SELECT AVG(salario)
                 FROM medicos);
```

```
-- Ejercicio 10
-- EXPLAIN
SELECT p.Nombre, p.FechaIngreso
FROM pacientes AS p
INNER JOIN medicos AS m ON p.IdMedico = m.id
WHERE m.Especialidad LIKE 'Cardiologia'
ORDER BY Especialidad, FechaIngreso;
```

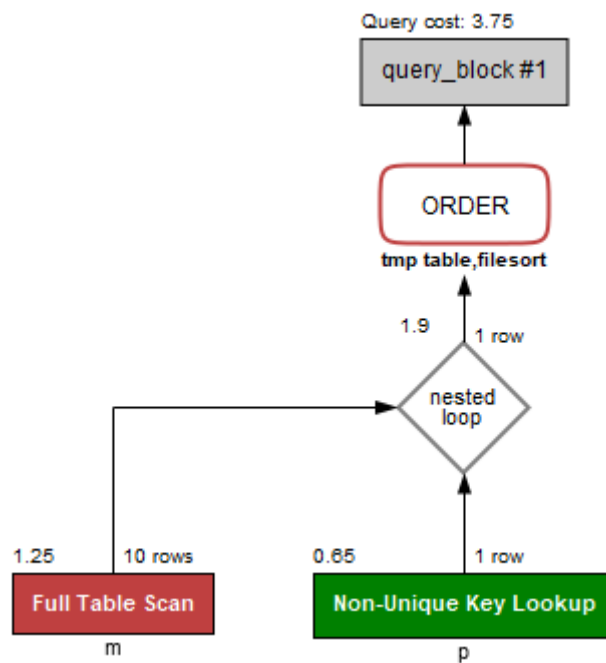
```
-- EXPLAIN
SELECT p.Nombre, p.FechaIngreso
FROM pacientes AS p
INNER JOIN medicos AS m ON p.IdMedico = m.id
WHERE m.Especialidad LIKE 'Cardiologia'
ORDER BY FechaIngreso;
```

10.

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	m	<u>NULL</u>	ALL	PRIMARY	<u>NULL</u>	<u>NULL</u>	<u>NULL</u>	10	11.11	Using where; Using temporary; Using filesort
	1	SIMPLE	p	<u>NULL</u>	ref	FK_pacientes_medicos	FK_pacientes_medicos	5	hospital.m.Id	1	100.00	<u>NULL</u>



	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	m	<u>NULL</u>	ALL	PRIMARY	<u>NULL</u>	<u>NULL</u>	<u>NULL</u>	10	11.11	Using where; Using temporary; Using filesort
	1	SIMPLE	p	<u>NULL</u>	ref	FK_pacientes_medicos	FK_pacientes_medicos	5	hospital.m.Id	1	100.00	<u>NULL</u>



En la primera e programa opta por la forma mas comoda que es el ordenarlo por la Especialidad, ya que usa como referencia una llave primaria y le es mas facil encontrar los medicos de la especialidad indicada.

De la otra forma se realiza el mismo proceso con la diferencia de que tiene que analizar todas las fechas y ordenarlas, en este caso de mas antiguas a mas recientes y este proceso le lleva mucho mas tiempo. Utiliza un Sort y esto le hace gastar 1,85 mas que en el anterior y hace que pase de 1.9 a 3.75. Por lo tanto si se busca la mejor optimizacion se recomendaria usar la primera manera que la segunda.