4. Dada la siguiente consulta SQL:

select c.primer\_nombre, c.apellidos

from pedidos p inner join clientes c on p.clienteID = c.clienteID

inner join detallepedidos d ON p.PedidoID = d.PedidoID

where year(c.fecha\_nacimiento) between 1980 and 1986 and genero = "F" and

c.ingresos\_anuales >= (

select MAX(ingresos\_anuales) AS `90th\_percentile`

from (

select ingresos\_anuales, PERCENT\_RANK() OVER (ORDER BY ingresos\_anuales)

AS percentile\_rank

from clientes

where year(fecha\_nacimiento) between 1970 and 1980 and genero = "F"

) AS ranked\_sales

where percentile\_rank <= 0.9

)

group by c.clienteID

having sum((precio\_unitario - descuento\_unitario) \* cantidad) >= (

select avg(total)

from (

select sum( (precio\_unitario - descuento\_unitario) \* cantidad) as total

from pedidos p inner join detallepedidos d ON p.PedidoID = d.PedidoID

where year(fecha\_venta) = 2021

group by clienteID

) as totales2021

);

1. Describir en lenguaje natural que realiza la consulta.

Esta consulta nos devuelve el primer apellido, el nombre de los clientes nacidos entre 1980 y 1986 cuyo genero sea ‘F’, sus ingresos anuales son mayores iguales que el máximo de los ingresos anuales de los clientes que han nacido entre 1970 y 1980 y el percent Rank es menor igual a 0.9

b. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.

c. Crear los índices que permitan optimizar el coste de las consultas, analizando plan de consulta y coste para cada uno de los casos, justificando que solución es la mejor. Cambia la consulta como veas necesario, pero el resultado de esta no debe cambiar.