**Nombre:** Pedro Antonio

**Apellidos:** Ruiz Martínez

**DNI:** 43125340H

**1. Pasadas las fiestas navideñas, Talleres Faber quiere enviar publicidad a sus clientes. Para ello se quiere seleccionar a aquéllos a los que no se les ha emitido ninguna factura desde el 6 de Enero. Obtén un listado con Nombre, Apellidos y Dirección de esos clientes.**

SELECT Nombre, Apellidos, Direccion

FROM clientes

WHERE CodCliente NOT IN (

SELECT facturas.CodCliente FROM facturas WHERE clientes.CodCliente = facturas.CodCliente AND FechaFACTURA>='2011-01-06');

**2.El gerente quiere conocer los apellidos y el nombre del empleado que más horas ha trabajado. La consulta debe mostrar 2 columnas:**

* **Los apellidos y el nombre, separados por una coma y un espacio con el encabezado ‘Apellidos y nombre’.**
* **Las horas totales trabajadas, con el encabezado ‘Horas totales trabajadas’.**

**SIN INNER JOIN**:

Select concat(empleados.Apellidos," ,", empleados.Nombre) as "Apellidos y nombre", sum(Horas) as "Horas totales trabajadas"

from empleados,intervienen

where empleados.CodEmpleado=intervienen.CodEmpleado

group by intervienen.CodEmpleado

order by sum(Horas) desc LIMIT 0,1;

**CON INNER JOIN**:

SELECT CONCAT(Apellidos,",", Nombre) AS "Apellidos y Nombre", SUM(horas) AS "Horas totales trabajadas"

FROM empleados

INNER JOIN intervienen ON empleados.CodEmpleado=intervienen.CodEmpleado

GROUP BY intervienen.CodEmpleado

ORDER BY SUM(horas) DESC LIMIT 0,1;

3**.¿Cuál es el importe total facturado este mes por Talleres Faber?**

**IMPORTE TOTAL ACTUACIONES:**

SELECT SUM(horas\*50) AS PRECIOREPARACION

from realizan;

**IMPORTE TOTAL RECAMBIOS:**

SELECT sum(PRECIOREFERENCIA\*UNIDADES) AS "Importe Total Recambios"

FROM recambios, incluyen

WHERE recambios.IdRecambio=incluyen.IdRecambio;

**CONTEO DE LAS FACTURAS:**

SELECT COUNT(IdFactura) AS "Número total de facturas emitidas"

FROM Facturas;

**4.** **Obtener un listado con el número de reparaciones en las que ha intervenido cada uno de los empleados. El listado debe mostrar:**

1. **El número de reparaciones, los apellidos y el nombre del empleado.**
2. **El número de reparaciones debe aparecer en orden descendente.**

SELECT Apellidos, nombre, COUNT(IdReparacion) AS Num\_Reparaciones

FROM empleados, intervienen

WHERE empleados.CodEmpleado = intervienen.CodEmpleado

GROUP BY empleados.CodEmpleado

ORDER BY Num\_Reparaciones DESC;

**5. Los socios del taller pretenden dar un servicio cada vez más rápido a sus clientes. Para ello han previsto que las reparaciones se realicen antes de 2 días. Interesa conocer las reparaciones que han superado esa previsión y los empleados que la han realizado. Las columnas que se deben mostrar son:**

1. **Nombre de los empleados, Fecha de entrada, Fecha de salida y Fecha de salida prevista (2 días a partir de la fecha de entrada).**
2. **Combinar las tablas utilizando la Combinación interna (INNER JOIN).**

SELECT Nombre, FechaEntrada, FechaSalida, DATE\_ADD(FechaEntrada, INTERVAL 2 DAY) AS "Fecha de salida prevista"

FROM empleados

INNER JOIN intervienen ON empleados.CodEmpleado=intervienen.CodEmpleado

INNER JOIN reparaciones ON intervienen.IdReparacion=reparaciones.IdReparacion

WHERE FechaSalida > DATE\_ADD(FechaEntrada, INTERVAL 2 DAY);

**6. La administración quiere comprobar que los únicos empleados que no han intervenido en reparaciones son aquéllos que se encargan de labores de organización y administración del taller. Hacer una consulta que devuelva los nombres y categorías de los empleados que no han intervenido en ninguna reparación.**

SELECT nombre, categoria

FROM empleados

WHERE CodEmpleado

NOT IN (SELECT intervienen.CodEmpleado

FROM intervienen

WHERE empleados.CodEmpleado=intervienen.CodEmpleado);

**7. Interesa conocer cuál es el vehículo con más kilómetros que ha visitado el taller. Mostrar los kilómetros, la marca, el modelo, el color y la matrícula.**

SELECT KM, MARCA, Modelo, Color, vehiculos.Matricula

FROM vehiculos, reparaciones

WHERE vehiculos.Matricula=reparaciones.Matricula

ORDER BY KM DESC LIMIT 0,1;

**8. En Talleres Faber están pensando en hacer más rentable la gestión de los recambios. Se pretende optimizar el stock, para ello se solicita un listado con los siguientes datos: IdRecambio, Descripción, UnidadBase y Stock, de los artículos que superen el stock considerado como óptimo, según los siguientes criterios:**

1. Para los artículos cuya unidad base se mide como**una unidad** se considera el stock óptimo hasta 10 artículos.
2. Para los artículos cuya unidad base **no sea una unidad** el stock considerado como óptimo es hasta 4 artículos.

SELECT IdRecambio, Descripcion, Unidadbase, Stock

FROM recambios

WHERE NOT unidadbase='unidad' AND stock>4 OR Unidadbase='unidad' AND stock>10;

**9. Con relación a las actuaciones, se considera importante saber cuántas reparaciones se realizan de importe superior a la media. La consulta nos devolverá un solo dato: el número de reparaciones realizadas cuyo importe supera el importe medio de las actuaciones.**

**Importe medios de las actuaciones:**

SELECT AVG(IMPORTE) AS "Importe medio de las actuaciones" FROM Actuaciones;

**Descripción e importe de las reparaciones que superan el importe medio:**

SELECT descripcion, importe

FROM actuaciones

WHERE importe>(SELECT AVG(importe) FROM actuaciones);

**Conteo total de reparaciones que superan el importe medio:**

SELECT COUNT(importe) as "Reparaciones que superan el importe medio"

FROM actuaciones

WHERE importe>(SELECT AVG(importe) FROM actuaciones);

**10. Obtener un listado que nos informe del número de facturas que hemos emitido a cada cliente. El listado tendrá dos columnas: el nombre completo del cliente y el número de facturas que le corresponden.**

SELECT CONCAT(Nombre," ", Apellidos) as "Nombre completo", COUNT(idfactura) as "Número de Facturas"

from clientes, facturas

WHERE facturas.CodCliente=clientes.CodCliente

GROUP BY facturas.codcliente

ORDER BY COUNT(idfactura) DESC LIMIT 0,1;