***Base de datos comercio.***

13. Mostrar el código de ticket, fecha y nombre del cliente sin que se pierda ningún ticket, aunque su cliente sea NULL.

**SELECT ticket.codigo, ticket.fecha, cliente.nombre FROM ticket LEFT JOIN cliente on cliente.codigo = ticket.cliente;**

14. Mostrar el nombre y dirección de los clientes junto con el nombre de su municipio sin perder aquellos municipios que no tienen clientes.

**SELECT cliente.nombre, cliente.direccion, municipio.nombre FROM cliente RIGHT JOIN municipio ON cliente.municipio = municipio.codigo;**

15. Escribe una consulta que devuelva el código de ticket, línea, código y descripción del artículo para todas las líneas del ticket 108, aunque el código de artículo sea nulo.

**SELECT ticket.codigo, linea\_ticket.nlinea, articulo.codigo, articulo.descripcion FROM ticket, articulo RIGHT JOIN linea\_ticket ON linea\_ticket.articulo = articulo.codigo WHERE ticket.codigo = 108;**

16. Si has resuelto la sentencia anterior con RIGHT JOIN, resuélvela con LEFT JOIN. Si la has resuelto con LEFT JOIN, resuélvela con RIGHT JOIN.

**SELECT ticket.codigo, linea\_ticket.nlinea, articulo.codigo, articulo.descripcion FROM ticket, linea\_ticket left JOIN articulo ON linea\_ticket.articulo = articulo.codigo WHERE ticket.codigo = 108;**

17. Mostrar el nombre de cada municipio y el número de clientes residiendo en cada uno de ellos. Los municipios que no tienen ningún cliente no aparecerán.

**SELECT municipio.nombre, COUNT(cliente.codigo) FROM cliente LEFT JOIN municipio ON cliente.municipio = municipio.codigo GROUP BY municipio.nombre;**

18. Mostrar el nombre de cada municipio y el número de clientes residiendo en cada uno de ellos. Los municipios que no tienen ningún cliente también aparecerán.

**SELECT municipio.nombre, COUNT(cliente.codigo) FROM cliente RIGHT JOIN municipio ON cliente.municipio = municipio.codigo GROUP BY municipio.nombre;**

19. Escribe una consulta que devuelva el código y fecha de cada ticket junto al nombre del vendedor que realizó la venta y el nombre del cliente destinatario del ticket. No se debe perder ningún ticket aunque no tenga código de vendedor o código de cliente.

**SELECT ticket.codigo, ticket.fecha, cliente.nombre, vendedor.nombre FROM ticket LEFT JOIN vendedor ON vendedor.codigo = ticket.vendedor LEFT JOIN cliente ON ticket.cliente = cliente.codigo;**

20. Obtén el nombre de los clientes y el número de tickets en los que aparece cada uno. También se mostrará el nombre de los clientes que no aparecen en ningún ticket.

**SELECT cliente.nombre, COUNT(ticket.codigo) FROM ticket LEFT JOIN cliente ON ticket.cliente = cliente.codigo GROUP BY cliente.nombre;**

21. Escribe una sentencia que muestre el código y nombre de cada vendedor y el total de ventas (importe total de sus tickets ignorando posibles descuentos e impuestos).

**select vendedor.codigo, vendedor.nombre, COUNT(linea\_ticket.cant)\* linea\_ticket.precio FROM ticket right jOIN linea\_ticket ON ticket.codigo = linea\_ticket.ticket LEFT JOIN vendedor ON ticket.vendedor = vendedor.codigo GROUP BY VENDEDOR;**

22. Escribe una sentencia que muestre el nombre de la provincia y la cantidad total de artículos vendidos a clientes que residen en dicha provincia.

**select provincia.nombre, count(ticket.CLIENTE)FROM TICKET LEFT JOIN CLIENTE ON ticket.cliente = cliente.codigo LEFT JOIN municipio on municipio.codigo= cliente.municipio left join provincia on provincia.codigo = municipio.provincia group by provincia;**

23. Escribe una consulta que muestre el código y nombre de aquellos clientes de la provincia de Castellón (con código de provincia ‘CA’) que en total han gastado más de 100 euros (ignora descuentos e impuestos).

**select cliente.codigo, cliente.nombre, provincia.nombre from linea\_ticket left join ticket on linea\_ticket.ticket = ticket.codigo left join cliente on ticket.cliente = cliente.codigo left join municipio on cliente.municipio = municipio.codigo left join provincia on provincia.codigo = municipio.provincia where provincia.nombre = "Castellón" and count(linea\_ticket.cant) \* linea\_ticket.precio >300;**

24. Escribe una consulta que obtenga el nombre de cada vendedor y el de su jefe.

***Base de datos instituto.***

25. Obtén el nombre de cada asignatura, así como el curso, edificio y aula donde se imparte.

**select asignatura.nombre, asignatura.curso, imparte.claseAula, imparte.edificioAula FROM asignatura, imparte;**

26. Obtén el nombre completo de cada profesor junto con el nombre y curso de cada una de las asignaturas asignatura que imparte.

**select profesor.nombre, asignatura.nombre, asignatura.curso from profesor left join ensenya on ensenya.profesor = profesor.dni left join asignatura on asignatura.codigo = ensenya.asignatura order by profesor.nombre ASC;**

27. Obtén el nombre y apellidos de los alumnos nacidos en 1999 junto con el nombre de cada una de las asignaturas que cursa y la nota obtenida.

**select alumno.nombre, alumno.apellido1, asignatura.nombre, asignatura.curso from alumno left join matricula on matricula.alumno = alumno.dni right join asignatura on matricula.asignatura = asignatura.codigo order by alumno.nombre ASc;**

28. Obtén el nombre de los profesores y el número de alumnos que tienen.

**select profesor.nombre, count(alumno.dni) from alumno Right join reparto on reparto.alumno = alumno.dni right join ensenya on ensenya.grupo = reparto.grupo right join profesor on ensenya.profesor = profesor.dni group by profesor;**

29. Para cada asignatura, obtén el nombre, el curso y la nota media obtenida por los alumnos que la cursan.

**SELECT asignatura.nombre, asignatura.curso, AVG(matricula.nota) FROM asignatura RIGHT JOIN matricula ON matricula.asignatura = asignatura.codigo GROUP BY asignatura.nombre;**

30. Obtén el nombre completo de cada alumno y la cantidad de asignaturas en que está matriculado cada uno de ellos siempre y cuando esté matriculado en menos de 5 asignaturas.

**SELECT alumno.nombre, alumno.apellido1, alumno.apellido2, COUNT(matricula.alumno) from alumno LEFT JOIN matricula ON alumno.dni = matricula.alumno WHERE COUN(matricula.alumno) > 5 GROUP BY alumno.nombre;**