

Sea la base de datos coches con las siguientes tablas:

MARCAS (codmarca, nombre, pais)

codmarca: Código de la marca los coches.

nombre: Nombre de la marca. No nulos.

pais: país de origen de la marca.

COCHES (codcoche, codmarca, modelo, matricula, color)

codcoche: Código del coche.

codmarca: Marca del coche.

modelo: Modelo del coche.

matricula: Matrícula del coche.

color: Color del coche. Valor por defecto: blanco.

FK (codmarca → marcas) No nulos.

CONCESIONARIOS (cifc, nombre, ciudad, cifcmatriz)

cifc: Cif del concesionario

nombre: Nombre del concesionario. No nulos.

ciudad: Ciudad donde está localizado el concesionario.

cifcmatriz: Código del concesionario matriz.

FK (cifcmatriz → concesionarios) Puede tomar valores Nulos.

CLIENTES (dni, nombre, apellidos, ciudad)

dni: Dni del cliente.

nombre: Nombre del cliente. No nulos.

apellidos: Apellidos del cliente. No nulos.

ciudad: Ciudad de origen del cliente.

DISTRIBUCION (cifc, codcoche, fecha)

cifc: Código del concesionario al que llega el coche.

codcoche: Código del coche distribuido.

fecha: Fecha en la que se distribuyó el coche al concesionario. Un coche puede pasar, hasta que es vendido, por varios concesionarios de coches. No nulos.

Valor por defecto: 1-1-1990.

FK1 (cifc → concesionarios) No nulos.

FK2 (codcoche → coches) No nulos.

VENTAS (cifc, codcoche, dni, fechaventa, pvp)

cifc: Código del concesionario que vendió el coche.

codcoche: Código del coche comprado.

dni: Cliente comprador del coche.

fechaventa: Fecha en la que se realiza la venta. No nulos.

pvp: Precio de venta del coche. No nulos. Valor por defecto: 0

FK1 (cifc → concesionarios) No nulos.

FK2 (dni → clientes) No nulos.

FK3 (codcoche → coches) No nulos.

1. Obtener la información de los coches ordenada alfabéticamente por nombre de marca.

```
SELECT marcas.nombre, coches.*  
FROM coches, marcas  
WHERE coches.codmarca=marcas.codmarca  
ORDER BY marcas.nombre;
```

2. Obtener un listado de los coches vendidos por más de 10000 euros.

```
SELECT coches.*  
FROM coches, ventas  
WHERE coches.codcoche=ventas.codcoche  
AND ventas.pvp>10000;
```

3. Obtener el total de vehículos distribuidos a cada concesionario.

```
SELECT cifc, COUNT(*)  
FROM distribucion  
GROUP BY cifc;
```

4. Obtener el precio de la venta más cara y el de la venta más barata.

```
SELECT MIN(pvp), MAX(pvp)  
FROM ventas;
```

5. Obtener la información de los clientes que han comprado algún coche en algún concesionario de su ciudad de origen.

```
SELECT DISTINCT clientes.*  
FROM clientes, concesionarios, ventas  
WHERE ventas.cifc=concesionarios.cifc  
AND ventas.dni=clientes.dni  
AND clientes.ciudad=concesionarios.ciudad;
```

6. Obtener los códigos de coches que hayan sido adquiridos por un cliente de Sevilla a un concesionario de Sevilla.

```
ELECT ventas.codcoche  
FROM clientes, concesionarios, ventas  
WHERE ventas.cifc=concesionarios.cifc  
AND ventas.dni=clientes.dni  
AND upper(concesionarios.ciudad)='SEVILLA'  
AND upper(clientes.ciudad)='SEVILLA';
```

7. Obtener el DNI de los clientes que han comprado algún coche a un concesionario de Madrid.

```
SELECT DISTINCT ventas.dni  
FROM concesionarios, ventas  
WHERE concesionarios.cifc=ventas.cifc  
AND upper(concesionarios.ciudad)='MADRID';
```

8. Obtener el nombre y apellido de los clientes cuyo dni es menor que el del cliente Juan López.

```
SELECT c.nombre, c.apellidos  
FROM clientes c  
WHERE c.dni<(SELECT c1.dni  
FROM clientes c1  
WHERE upper(c1.nombre)='JUAN' AND upper(c1.apellidos)='LOPEZ');
```

9. Obtener el nombre y el apellido de los clientes cuyo dni es menor que el de los clientes de Huelva.

```
SELECT c.nombre, c.apellidos
FROM clientes c
WHERE c.dni < ALL(SELECT c1.dni
FROM clientes c1
WHERE upper(ciudad)='HUELVA');
```

10. Obtener el nombre y los apellidos de los clientes que han comprado como mínimo un coche blanco y otro plata.

```
SELECT clientes.nombre, clientes.apellidos
FROM clientes
WHERE clientes.dni IN (SELECT ventas.dni
FROM ventas, coches
WHERE ventas.codcoche=coches.codcoche
AND upper(coches.color)='BLANCO')
AND clientes.dni IN (SELECT ventas.dni
FROM ventas, coches
WHERE ventas.codcoche=coches.codcoche
AND upper(coches.color)='PLATA');
```

11. Obtener el dni de los clientes cuya ciudad sea la última de la lista alfabética de las ciudades donde hay concesionarios.

```
SELECT dni
FROM clientes
WHERE ciudad=(SELECT MAX(ciudad)
FROM concesionarios);
```

12. Obtener el código de los coches vendidos por algún concesionario de Madrid.

```
SELECT DISTINCT ventas.codcoche
FROM ventas, concesionarios
WHERE ventas.cifc=concesionarios.cifc AND upper(concesionarios.ciudad)='MADRID';
```

13. Obtener los dni de los clientes que sólo han comprado coches al concesionario 3.

```
SELECT DISTINCT ventas.dni
FROM ventas
WHERE ventas.dni NOT IN (SELECT ventas.dni
FROM ventas
WHERE ventas.cifc!=3);
```

14. Obtener el cifc de los concesionarios que han vendido la misma marca de coche a todos los clientes.

```
SELECT ventas.cifc
FROM ventas, coches
WHERE ventas.codcoche=coches.codcoche
GROUP BY ventas.cifc
HAVING COUNT (DISTINCT coches.codmarca)=1;
```

15. Obtener el nombre de las marcas para las que se haya vendido algún coche.

```
SELECT DISTINCT marcas.nombre
FROM marcas, coches, ventas
WHERE ventas.codcoche=coches.codcoche
AND coches.codmarca=marcas.codmarca;
```

16. Obtener el nombre de los concesionarios en los que se haya distribuido más de dos coches.

```
SELECT concesionarios.nombre
FROM concesionarios, distribucion
WHERE concesionarios.cifc=distribucion.cifc
GROUP BY distribucion.cifc, concesionarios.nombre
HAVING COUNT(*)>2;
```

17. Obtener el total pagado por vehículos en concesionarios de la ciudad de Sevilla.

```
SELECT SUM(ventas.pvp)
FROM ventas, concesionarios
WHERE ventas.cifc=coches.cifc AND
upper(coches.ciudad)='SEVILLA';
```

18. Obtener aquellos coches que no se hayan distribuido o que no se hayan vendido todavía.

```
SELECT coches.codcoche
FROM coches
WHERE coches.codcoche NOT IN (SELECT DISTINCT distribucion.codcoche
                              FROM distribucion)
OR
coches.codcoche NOT IN (SELECT DISTINCT ventas.codcoche
                        FROM ventas);
```

19. Obtener un listado con la información de todos los clientes que no hayan comprado ningún coche.

```
SELECT clientes.*
FROM clientes
WHERE clientes.dni NOT IN (SELECT ventas.dni FROM ventas);
```

20. Obtener un listado con las ventas de coches posteriores al año 2000 que se hayan realizado en la ciudad de Sevilla o de Málaga.

```
SELECT ventas.*
FROM ventas, concesionarios
WHERE ventas.cifc=concesionarios.cifc AND
extract(year FROM ventas.fechaventa)>2000 AND
(upper(concesionarios.ciudad)='SEVILLA' OR upper(concesionarios.ciudad)='MALAGA');
```

21. Obtener un listado del número total de coches vendidos por marca ordenado alfabéticamente.

```
SELECT marcas.nombre, COUNT(*)
FROM coches, ventas, marcas
WHERE coches.codcoche=ventas.codcoche
AND marcas.codmarca=coches.codmarca
GROUP BY marcas.codmarca, marcas.nombre
ORDER BY marcas.nombre;
```

22. Obtener un listado con el número total de coches comprados por cada cliente.

```
SELECT dni, COUNT(*)  
FROM ventas  
GROUP BY dni;
```

23. Obtener los concesionarios para los que se haya vendido al menos dos coches de la marca SEAT.

```
SELECT ventas.cifc  
FROM ventas, coches, marcas  
WHERE ventas.codcoche=coches.codcoche  
AND coches.codmarca=marcas.codmarca  
AND upper(marcas.nombre)='SEAT'  
GROUP BY ventas.cifc  
HAVING COUNT(ventas.codcoche)>=2;
```

26. Obtener un listado con la información de todas las ventas de coches con el nombre del comprador y nombre del concesionario donde se realizó la venta.

```
SELECT clientes.nombre, concesionarios.nombre, ventas.*  
FROM clientes, concesionarios, ventas  
WHERE ventas.dni=clientes.dni  
AND ventas.cifc=concesionarios.cifc;
```

27. Obtener la información de los clientes que han comprado algún coche en algún concesionario de fuera de su ciudad de origen.

```
SELECT DISTINCT clientes.*  
FROM clientes, ventas, concesionarios  
WHERE ventas.dni=clientes.dni AND ventas.cifc=concesionarios.cifc  
AND clientes.ciudad!=concesionarios.ciudad;
```

28. Obtener un listado con la información de los concesionarios a los que se les ha distribuido un coche burdeos y todavía no lo hayan vendido.

```
SELECT concesionarios.*  
FROM concesionarios  
WHERE concesionarios.cifc = (SELECT distribucion.cifc  
FROM distribucion, coches  
WHERE distribucion.codcoche=coches.codcoche  
AND upper(coches.color)='BURDEOS'  
AND coches.codcoche  
NOT IN (SELECT ventas.codcoche  
FROM ventas));
```

29. Obtener todas las ventas de vehículos en las que el comprador y el concesionario no estén en la misma ciudad.

```
SELECT ventas.*
FROM ventas, concesionarios, clientes
WHERE ventas.cifc=concesionarios.cifc
AND ventas.dni=clientes.dni
AND upper(concesionarios.ciudad)<>upper(clientes.ciudad);
```

30. Escribir una sentencia SQL que permita insertar un nuevo coche de la marca FERRARI. El resto de los campos quedan a la elección del alumno.

Es necesario dar de alta la marca FERRARI:

```
INSERT INTO marcas (codmarca, nombre)
VALUES (18, 'FERRARI');
```

```
INSERT INTO coches (codcoche, codmarca, modelo, matricula)
VALUES (11, 18, 'ElMejor', '1234A');
```

31. Escribir una sentencia SQL que permita subir el pvp de todas las operaciones de venta de coches en un 10%.

```
UPDATE ventas
SET pvp = pvp*1.1;
```

32. Escribir una sentencia SQL que permita añadir una columna teléfono a la tabla de clientes de tipo numérico de 12. Puede tomar valores nulos.

```
ALTER TABLE clientes
ADD telefono NUMBER(12);
```

33. Escribir una sentencia SQL para eliminar todas las filas de la tabla COCHES. Cuando la ejecutáis, ¿qué errores genera Oracle y a qué se deben? ¿Cuáles son las posibles soluciones a estos errores?

```
DELETE FROM coches;
```

-Errores son de porque hay una FK asociada a esta tabla.

-Desactiva la FK y borra.

34. Escribir una sentencia SQL que permita desactivar las restricciones de integridad referencial de la tabla ventas.

```
ALTER TABLE ventas
DISABLE CONSTRAINT venfk1
DISABLE CONSTRAINT venfk2
DISABLE CONSTRAINT venfk3;
```

35. Escribir una sentencia SQL que permita eliminar la columna teléfono de la tabla clientes.

```
ALTER TABLE clientes
DROP COLUMN telefono;
```

36. Añade una nueva columna denominada fechaAlta a la tabla marcas.

```
ALTER TABLE marcas  
ADD fechaAlta DATE;
```

37. Modifica la columna fechaAlta de la tabla marcas para que tome la fecha actual del sistema.

```
ALTER TABLE marcas  
MODIFY fechaAlta DATE DEFAULT CURRENT_DATE;
```

38. Borra todos los concesionarios para los que no haya coches distribuidos.

```
DELETE FROM concesionarios  
WHERE NOT EXISTS (SELECT *  
FROM distribucion  
WHERE distribucion.cifc=concesionarios.cifc);
```

39. Borra todos los clientes que hayan comprado menos de 2 coches.

```
DELETE FROM clientes  
WHERE 2 > (SELECT COUNT(*)  
FROM ventas  
WHERE ventas.dni=clientes.dni);
```

40. Construir una nueva tabla cli2 conteniendo una lista de los dni y nombres de los clientes que han comprado algún FIAT.

```
CREATE TABLE cli2 AS (SELECT clientes.dni, clientes.nombre  
FROM clientes, ventas, coches, marcas  
WHERE ventas.dni=clientes.dni  
AND ventas.codcoche=coches.codcoche  
AND marcas.codmarca=coches.codmarca  
AND upper(marcas.nombre)='FIAT');
```

41. Construir una nueva tabla coc2 conteniendo una lista con la marca, modelo y matrícula de todos los coches blancos.

```
CREATE TABLE coc2 AS (SELECT codmarca, modelo, matricula  
FROM coches  
WHERE upper(color)='BLANCO');
```

43. Escribir una sentencia SQL que permita habilitar todas las restricciones de integridad referencial de la tabla ventas.

```
ALTER TABLE ventas  
ENABLE CONSTRAINT venfk1  
ENABLE CONSTRAINT venfk3;
```

45. Añadir la restricción UNIQUE a la combinación de campo nombre y apellidos de la tabla clientes.

```
ALTER TABLE clientes  
ADD CONSTRAINT NomAp_Unique UNIQUE(nombre, apellidos);
```

46. Asegurar que el campo ciudad de la tabla concesionarios no puede tomar valores nulos.

```
ALTER TABLE concesionarios  
MODIFY ciudad NOT NULL;
```