

## Notas útiles para resolver los ejercicios

- El formato que se utiliza en los ejercicios para describir las tablas del modelo relacional es el siguiente:

**NOMBRE\_TABLA** (**campo1**, **campo2** ...).

➔ Los campos que son **clave primaria** (PK, primary key) aparecen en negrita y subrayado.

➔ Los campos que son **claves externas** (FK, foreign key) aparecen en negrita. Además se aporta información adicional (tabla a la que referencia)

- **Formato de fecha:** para ver el formato admitido de fecha puedes ver la fecha actual con la instrucción:

```
select sysdate from dual;
```

Para modificar el formato de fecha admitido puedes utilizar la instrucción (modificando el formato a tu conveniencia):

```
Alter session set nls_date_format='DD-MM-YYYY';
```

- Comandos básicos LMD para manipular datos:

Visualiza todos los datos de una tabla:

```
select * from nombre_tabla;
```

Insertar datos en una tabla (todos los campos del registro):

```
insert into nombre_tabla values (valor1, valor2...)
```

Insertar datos en una tabla (no todos los campos del registro):

```
insert into nombre_tabla(campo1,campo3...) values (valor1, valor3...)
```

### Ejercicios: crear tablas sencillas

1. Crea las siguientes tablas.

a) REVISTA (codigo, sala, estanteria)

- Código es un número de 4 dígitos
- Sala es un nombre (máximo 10 caracteres de longitud)
- Estantería es un número de 1 dígito.

Comprueba que se ha creado bien con DESCRIBE.

b) ACTOR (DNI, nombre\_artístico, fecha\_nac)

- DNI es un texto con letras y números de 10 caracteres
- Nombre\_artístico es un texto de 20 caracteres como máximo
- Fecha\_nac es la fecha de nacimiento

Comprueba que se ha creado bien con DESCRIBE

c) Borra las tablas creadas (comando DROP).

### Ejercicios: crear tablas. Campos numéricos con precisión y escala

2. Crea la siguiente tabla, en la que se almacenarán datos de muebles.

MUEBLE (cod, alto, ancho, fondo, precio, stock)

- Cod es un texto de 3 caracteres
- Alto, ancho, fondo, precio y stock son números
- El alto, ancho y fondo permiten 2 dígitos enteros y 2 decimales
- El precio permite 3 dígitos enteros y 2 decimales
- El stock permite 4 dígitos sin decimales.

Inserta los siguientes datos. Si alguno da algún error indica dónde está el problema.

Mueble 1:      cod=12R  
                  Alto=60,03  
                  Ancho=25,30  
                  Fondo=50,05  
                  Precio= 102,99  
                  Stock=124

Mueble 2:      cod=03S  
                  Alto=110,00  
                  Ancho=10,55  
                  Fondo=33,10  
                  Precio= 78,99  
                  Stock=35

3. Crea la siguiente tabla, para guardar artículos de una tienda:

PRECIO\_ART (articulo, valor, fecha)

```
create table precio_art(
    articulo varchar2(10) primary key,
    valor number(5,-2),
    fecha date);
```

Inserta los siguientes datos. Si alguno da algún error indica dónde está el problema (ver [nota formato de fechas](#))

Registro 1: articulo = DELGER  
 Valor = 10565  
 Fecha = 23 de marzo de 2017 (modifica el formato)

Registro 2: articulo = SYNTR  
 Valor = 100,72  
 Fecha = 10 de enero de 2020 (modifica el formato)

Comprueba cómo se han guardado los datos en la tabla mediante la instrucción:

```
select * from precio;
```

¿Se ha modificado algún valor? ¿por qué?

4. Se ejecuta una serie de instrucciones SQL en una base de datos Oracle. A continuación se muestra, para cada instrucción, la respuesta que proporciona la base de datos.

```
SQL> insert into pr values ('aad3', 5.6, '2');
1 fila creada.
```

```
SQL> insert into pr values ('3founc5bayt', 67.54, 's');
Informe de error -
ORA-12899: el valor es demasiado grande para la columna "ALUMNO"."PR"."CAMP01" (real:
11, máximo: 10)
```

```
SQL> insert into pr values ('polk6hygtf', 5467.98, 'd');
Informe de error -
ORA-01438: valor mayor que el que permite la precisión especificada para esta columna
```

```
SQL> insert into pr values ('polk6hygtf', 6.154, 'e');
1 fila creada.
```

```
SQL> insert into pr values ('pol', 5467.98, 'dr');
Informe de error -
ORA-01438: valor mayor que el que permite la precisión especificada para esta columna
```

```
SQL> insert into pr values ('polk6hygtf', 6.154, 'dr');
Informe de error -
ORA-12899: el valor es demasiado grande para la columna "ALUMNO"."PR"."CAMP03" (real:
2, máximo: 1)
```

```
SQL> insert into pr values ('polrgt', 67.93, 'A');
Informe de error -
ORA-01438: valor mayor que el que permite la precisión especificada para esta columna

insert into pr values ('abcde', 2.93, 2);
Informe de error -
ORA-00001: restricción única (ALUMNO.SYS_C007345) violada

SQL> select * from pr;
CAMPO1          CAMPO2 C
-----
aad3              5,6 2
polk6hygtf       6,154 e
```

Interpreta esas instrucciones y escribe el comando necesario para crear la tabla PR.

### Ejercicios: crear tablas con relaciones (PK, FK y not null)

5. Se diseñan las siguientes tablas relacionadas:

PISO (**ref**, **dni\_propietario**, fecha\_compra)  
 dni\_propietario es FK, referencia a PROPIETARIO (piso)  
 fecha de compra es un campo obligatorio

PROPIETARIO (**dni**)

a) Interpreta el significado de los campos y sus relaciones. Crea las tablas empleando el formato de restricción de columna para todas las restricciones.

b) Inserta los siguientes datos:

- El propietario con DNI 85000000J tiene el piso de referencia 4501X, comprado el 29-03-2009
- El propietario con DNI 46000000N tiene 3 pisos, con referencias 1001A, 2120B y 1002V, comprados todos el día 10-02-2014
- El piso de referencia 7575F es propiedad de 98000000H desde hoy (sysdate)

c) ¿Puede un piso pertenecer a dos propietarios?

6. Se diseña la siguiente tabla:

PERSONA (**dni**, nombre, edad, **padre**)  
 padre es FK, referencia a PERSONA (dni)

a) Interpreta el significado de los campos y sus relaciones. Crea la tabla empleando el formato de restricción de columna para todas las restricciones.

Observa que hay una referencia (FK) a la misma tabla. Eso significa que solo podrán ponerse como padres en el campo "padre" personas que previamente estén dadas de alta en la tabla (campo "dni").

PERSONA

<u>dni</u>	nombre	edad	padre

Hay que tener en cuenta que:

- Dni es un texto de 9 caracteres
- Nombre es un texto de 20 caracteres
- Edad es un número de 3 dígitos

b) Introduce los datos:

- Juan, con DNI 10000000A y 32 años
- Luis, con DNI 20000000A (54 años) y su hijo Luis con DNI 40000000A (18 años)
- Daniel con DNI 30000000A (73 años) y su hijo Alberto con DNI 50000000A (51 años) y su nieta Eva con DNI 60000000A (19 años).

**Nota:** para insertar una persona que no tenga padre, se puede poner en el campo padre el valor null, que es equivalente a vacío.

7. Se quiere una base de datos en la que se almacenen datos de trabajadores. Para cada trabajador interesa conocer su nombre, nif, apellidos, edad, teléfono y puesto.

Hay que tener en cuenta que un mismo trabajador puede tener varios teléfonos, pero no se conoce a priori cuántos (podría tener uno, dos, tres, cuatro... o ninguno).

- a) Crea dos tablas relacionadas: TRABAJADOR y TLFNO\_TRABAJADOR. Elige los campos que debe tener cada una para cumplir los requisitos.

Considera que un teléfono solo puede ser de un trabajador (dos trabajadores no pueden tener el mismo teléfono). Recuerda que en el modelo relacional no puede haber campos multivaluados (una columna con más de un valor). Intenta que haya la menor información duplicada posible, para optimizar la base de datos.

Elige también el tipo de datos necesario para poder introducir los siguientes trabajadores, e introdúcelos en las tablas:

- Técnico de mantenimiento: Rubén Ruiz, de 33 años, con NIF 46000000W y teléfono 612000003.
- Conductora: Ana Isabel Mateos, de 42 años, con NIF 35000000T y teléfonos 916780001 y 610500001.

- b) ¿Qué cambios habría que hacer en las tablas para que dos empleados pudiesen tener el mismo teléfono? Borra las tablas y créalas de nuevo con la nueva condición.

Inserta a dos trabajadores que viven en el mismo domicilio y comparten teléfono fijo:

- Técnico de mantenimiento: Rubén Ruiz, de 33 años, con NIF 46000000W. Teléfono 916725899 y 612000003.
- Conductora: Ana Isabel Mateos, de 42 años, con NIF 35000000T. Teléfono 916725899 y teléfono 610500001.

### Ejercicios: crear tablas con restricciones

8. Se define la tabla:

MONTE(nombre, provincia, altura)

a) Crea la tabla, teniendo en cuenta:

- Sólo se podrán introducir montes más altos de 2000 metros.
- La altura es un campo obligatorio.

Crea todas las restricciones como restricciones de tabla.

b) Introduce los datos:

- Mulhacén 3479 metros. Granada.
- Aneto 3404 metros. Huesca
- Teide 3718 metros. Santa Cruz de Tenerife.
- Naranjo de Bulnes 2519 metros. Asturias

c) Comprueba que al intentar insertar los siguientes datos, todos dan error.

- Mulhacén 3480 metros. Granada.
- Puig Major 1445 metros. Islas Baleares
- Almanzor. Ávila.

9. Se tiene la siguiente tabla definida:

EMPRESARIO (nif, empresa, nombre, mail)

a) **Antes de crear la tabla**, para los siguientes datos, marca cuales podrán insertarse y cuáles no (en el orden en que aparecen). Indica la causa. Cuando no aparece un dato significa que no se conoce (no es un espacio en blanco).

	NIF	Empresa	Nombre	mail
1	85000000H	Gesco SL	Juan Colindres	director@gesco.es
2	44000000C	BotCO	Julia Martín	info@bot.com
3	13000000F	Tamar	Ana Palacios	informacion-tamar
4	77000000G	Tintor SA		info12345@gmail.com
5	75000000G	Next SA	Julia Martín	const-77@yahoo.es
6	99000000D	Gesco SL	Adrián Palomares	subdir@gesco.es
7	47000000S		Sergio Olivares	serol_89@google.com
8	62000000W	Alian	Jose María Lara	const-77@yahoo.es
9	54000000Q	DisCoTK	Agustín Moya	

b) Crea la tabla. Hay que tener en cuenta lo siguiente:

- No puede repetirse el nombre de empresa para dos empresarios.
- No puede repetirse el mail para dos empresarios.
- El mail debe contener el carácter @
- La empresa debe indicarse obligatoriamente para un empresario.

Crea todas las restricciones con el formato de restricción de columna.

c) Inserta los datos del apartado a) comprobando si fallan los que habías señalado

10. Se definen las siguientes tablas para crear una base de datos para una librería.

LIBRO (isbn, título, temática, posicion)

AUTOR (isbn, nombre)

FK: isbn referencia LIBRO(isbn)

a) En primer lugar, interpreta las tablas y, **sin crear las tablas**, responde a las preguntas:

¿Cuántos autores puede tener un libro?

¿Puede un autor haber escrito más de un libro?

¿Pueden estar libros escritos por el mismo autor en estantes distintos?

b) Crea las tablas teniendo en cuenta:

- isbn es varchar2(10)
- titulo y nombre son varchar2(30)
- temática es varchar2(20)
- posición es varchar2(5)
- Las temáticas posibles son 'terror', 'aventuras' e 'infantil'.
- El título del libro es un campo obligatorio.
- La posición se refiere a la estantería donde está el libro, y la forma de nombrarla es 'EST-x', siendo x un número o letra identificativa (1 carácter). Por ejemplo: EST-1, EST-2, EST-a, EST-H, etc

Las restricciones de tipo "check" y "foreign key" deben crearse con el formato de restricción de tabla. El resto pueden crearse como se prefiera.

c) Una vez creadas las tablas, comprueba el funcionamiento de las restricciones intentando insertar datos que no las cumplan. Debes construir un comando distinto que falle para cada restricción.

### Ejercicios: crear tablas con nombre de restricción

11. Se quiere implementar una base de datos para almacenar los extintores que tienen vagones de tren.

Se tienen una serie de vagones. Un vagón se identifica por el código del tren al que pertenece y el orden que ocupa dentro del él. Se conoce además su modelo y año de fabricación.

Se tienen una serie de extintores. Cada extintor tiene un número de referencia unívoco que lo identifica, e interesa conocer en qué vagón concreto está ubicado.

En un mismo vagón puede haber 0 extintores, 1 extintor, o más de 1.

Este es el diseño de las tablas:

VAGÓN (codigo\_tren, orden, modelo, año\_fab)

EXTINTOR (num\_ref, codigo\_tren, orden)

FK: (codigo\_tren,orden) referencia a VAGÓN (codigo\_tren, orden)

a) Crea las tablas teniendo en cuenta:

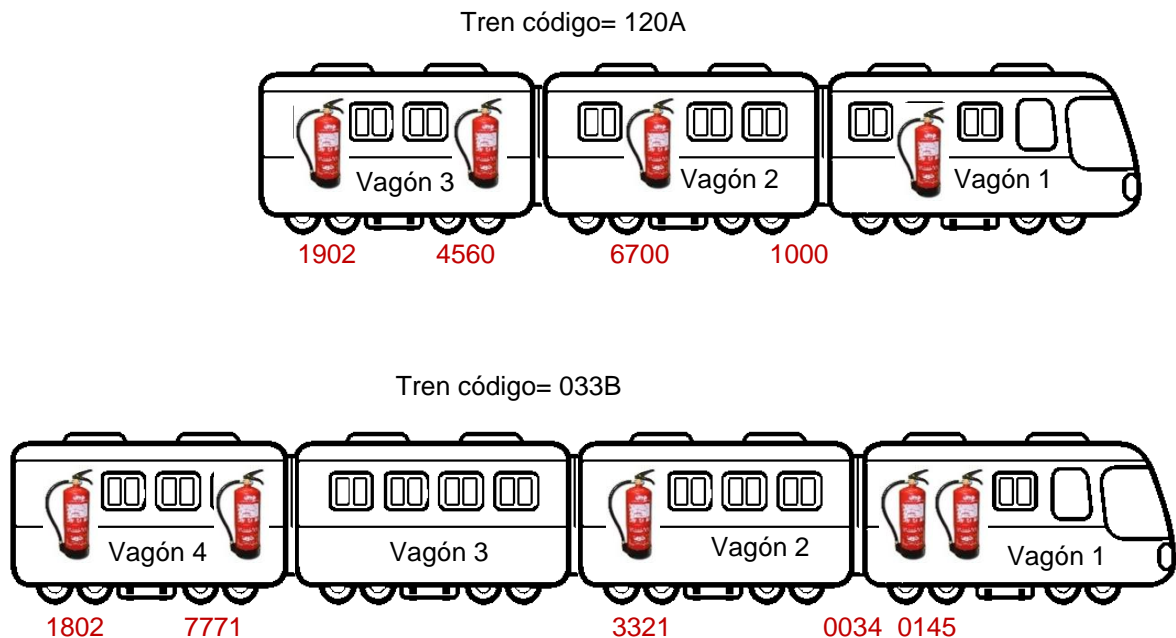
- El código de tren es un texto de 10 caracteres
- El orden del vagón es un número de 2 dígitos. Debe estar comprendido entre el 1 y el 20.
- El modelo del vagón es un texto de 10 caracteres
- El año de fabricación del vagón es un número de 4 dígitos. Siempre debe ser posterior al 2015.

Crea todas las restricciones con nombre de restricción (a tu elección).

b) Revisa las vistas de las constraints y comprueba que se han creado correctamente.

c) Inserta los datos del siguiente esquema. Ten en cuenta:

- Los números en rojo se refieren a los extintores.
- Los vagones 1 y 3 del tren 120A son del modelo 'D-2866 LUE'. El resto de vagones son del modelo 'H-333 DUE'.
- Todos los vagones se fabricaron en 2020, excepto las cabeceras (vagones de orden 1), que se fabricaron en 2018.



d) Intenta insertar los siguientes datos, que deben fallar. Comprueba el mensaje de error que da el fallo y busca en él el nombre de la restricción que se ha violado.

Vagón: tren '120A', orden 21, modelo 'D-2866 LUE', año de fabricación 2020

Vagón: tren '120A', orden 4, modelo 'D-2866 LUE', año de fabricación 2015

Vagón: tren '120A', orden 3, modelo 'D-2866 LUE', año de fabricación 2020

Extintor: número de referencia 1902. Se encuentra en el vagón del tren '120A', orden 3

Extintor: número de referencia 1903. Se encuentra en el vagón del tren '120A', orden 4



## Ejercicios: crear tablas con nombre de restricción y default

12. Se quiere almacenar información sobre los productos vendidos por una empresa de tornillos.

La empresa dispone de distintos vendedores, que se almacenan en la tabla Vendedor. Cada uno de los vendedores vende en una tienda en concreto.

Las tiendas se guardan en la tabla Tienda.

Los productos que se venden son tornillos, que se guardan en la tabla Tornillo.

Las ventas que hacen los vendedores se guardan en la tabla Venta. En esa tabla se refleja el vendedor, el producto vendido y en qué cantidad, y también la fecha de la venta.

Este es el diseño de la base de datos:

Tornillo (codigo, diametro)

Vendedor (dni, calle, ciudad)

FK: calle, ciudad referencia a Ciudad(calle, ciudad)

Tienda (calle, ciudad, comunidad)

Venta (vendedor, cod\_tornillo, diametro, fecha, cantidad)

FK: vendedor referencia vendedor(dni)

FK: cod\_tornillo, diametro referencia a tornillo (código, diámetro)

- a) En primer lugar, interpreta las tablas y, **sin crear las tablas**, responde a las preguntas:

¿Pueden existir distintos diámetros de un mismo código de tornillo?

¿Puede un vendedor vender distintos tornillos el mismo día?

¿Pueden tener dos tiendas que están en distinta ciudad el mismo nombre de calle?

¿por qué se incluye en la tabla Vendedor los campos “calle” y “ciudad” ?

¿Puede haber más de un vendedor que venda en la misma tienda?

- b) Crea las tablas. Debes incluir, aparte de las restricciones de clave primaria y externa, las siguientes:

- Los códigos de producto que vende la empresa son: AA1, AA2, AA3, AA4 y AA5. No puede haber ninguno más.
- El diámetro de un tornillo debe ser menor de 90.
- La cantidad de producto vendidos debe ser 5 como mínimo y 50 como máximo.
- La dirección de la venta debe empezar por ‘C’ o por ‘P’ obligatoriamente (corresponde a Calle y Plaza, respectivamente)
- La comunidad de una tienda es un campo obligatorio.
- Solo puede haber una tienda por comunidad autónoma: el campo “comunidad” no se puede repetir en la tabla Tienda.
- Si no se indica la fecha en una venta, la fecha por defecto será el día de hoy (sysdate).
- Si no se indica la cantidad en una venta, la cantidad por defecto será 25.

Para determinar el tipo de campo en cada caso, puedes consultar el apartado b) y revisar los datos que se van a insertar.

Las restricciones deben crearse con formato de columna siempre que sea posible.

Todas las restricciones de tipo “check” y “unique” deben tener nombre de restricción (a tu elección).

- c) Revisa las vistas de las constraints y comprueba que se han creado correctamente.
- d) Inserta las siguientes ventas. Ten en cuenta que para hacer inserciones en la tabla Venta previamente tienes que insertar vendedores, tiendas y productos.

DNI del vendedor	Dirección de la venta	Ciudad de la venta	Comunidad de la venta	Producto vendido (código y diámetro en mm)	Cantidad de productos vendidos	Fecha venta
10000000F	C. Jorge Juan	Madrid	Madrid	AA1, 20	30	12-2-21
50000000U	C. Pradillo	Oviedo	Asturias	AA5, 15	10	12-2-21
90000000T	C. Jorge Juan	Granada	Andalucía	AA2, 20	12	13-5-21
70000000Y	Plaza Mayor	Salamanca	Castilla León	AA3, 10	8	20-5-21
10000000F	C. Jorge Juan	Madrid	Madrid	AA2, 10	5	18-4-21
10000000F	C. Jorge Juan	Madrid	Madrid	AA2, 20	7	20-4-21

- e) Inserta la siguiente venta, en la que no se indica cantidad ni fecha. Comprueba que funcionan correctamente los valores por defecto.

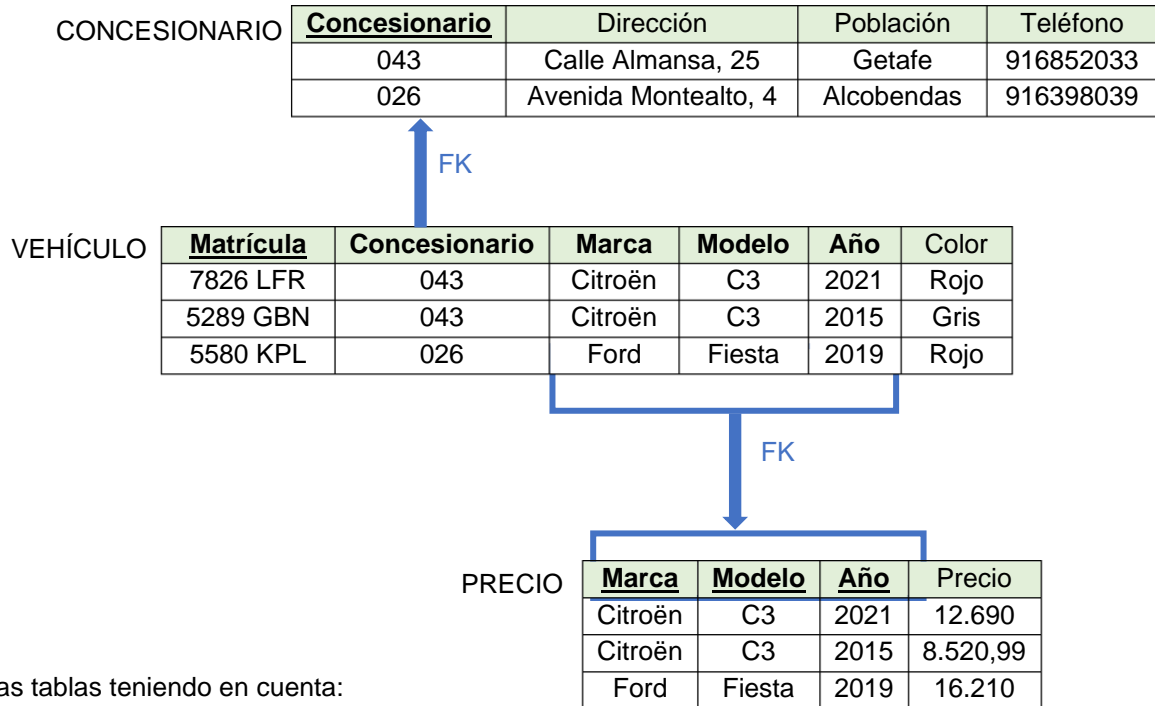
DNI del vendedor	Dirección de la venta	Ciudad de la venta	Comunidad de la venta	Producto vendido (código y diámetro en mm)	Cantidad de productos vendidos	Fecha venta
10000000F	C. Jorge Juan	Madrid	Madrid	AA1, 20		

- f) Intenta insertar los siguientes datos. Si las tablas están creadas correctamente, todos ellos deben fallar. Indica la causa del fallo en cada caso y revisa el mensaje de error que aparece.

- El vendedor 10000000F vende 7 tornillos 'AA1' de diámetro 20 el 12-2-21
- El vendedor 50000000U vende 10 tornillos 'AA3' con diámetro 20 el 12-12-21
- El vendedor 50000000U vende 2 tornillos 'AA7' con diámetro 20 el 12-11-21
- El vendedor 90000000T vende 54 tornillos 'AA1' con diámetro 20 el 1-5-21
- Se da de alta una tienda en la "Plaza Mayor" de Madrid, comunidad de Madrid.
- Se da de alta una tienda en la "Plaza Mayor" de la ciudad Riofrio sin indicar comunidad (null).
- Se da de alta el tornillo 'AA6' con diámetro 95.

**Ejercicios: crear tablas. Renombrar, borrado y truncado.**

13. Se tiene una base de datos de una empresa comercializadora de vehículos, compuesta por tres tablas. Analiza las tablas, sus claves y las relaciones entre ellas.



a) Crea las tablas teniendo en cuenta:

- Selecciona los tipo de datos teniendo en cuenta los datos de las tablas, y las posibles inserciones futuras.
- La matrícula debe tener un espacio intercalado.
- Si no se indica año, se considerará 2021.
- El precio es un número que debe admitir 6 dígitos enteros y 2 decimales.
- El precio debe ser mayor de 5.000
- El color debe solo puede ser Rojo, Azul, Gris, Blanco y Negro.

Crea las restricciones con formato de columna siempre que sea posible. Las de tipo “foreign key” deben tener nombre de restricción.

b) Inserta los datos que aparecen en las tablas.

El formato de los números decimales en la base de datos por defecto se introduce con un punto para la separación de los decimales y sin punto de los miles (es configurable). Por ejemplo 1.525,99 se introduciría como 1525.99.

c) Intenta truncar la tabla “concesionario” reservando el espacio y explica qué ocurre.

d) Cambia el nombre a la tabla “precio”. El nuevo nombre será “precio\_vehiculo” .

e) La tabla “vehiculo” referencia a la tabla “precio\_vehiculo”. Comprueba en las vistas de las constraints que la referencia entre ambas tablas se ha actualizado correctamente con el nuevo nombre.

f) Utiliza el siguiente comando para borrar la tabla “concesionario”:

```
drop table concesionario;
```

Explica qué ocurre y qué solución habría si se quisiera borrar esa tabla a toda costa.