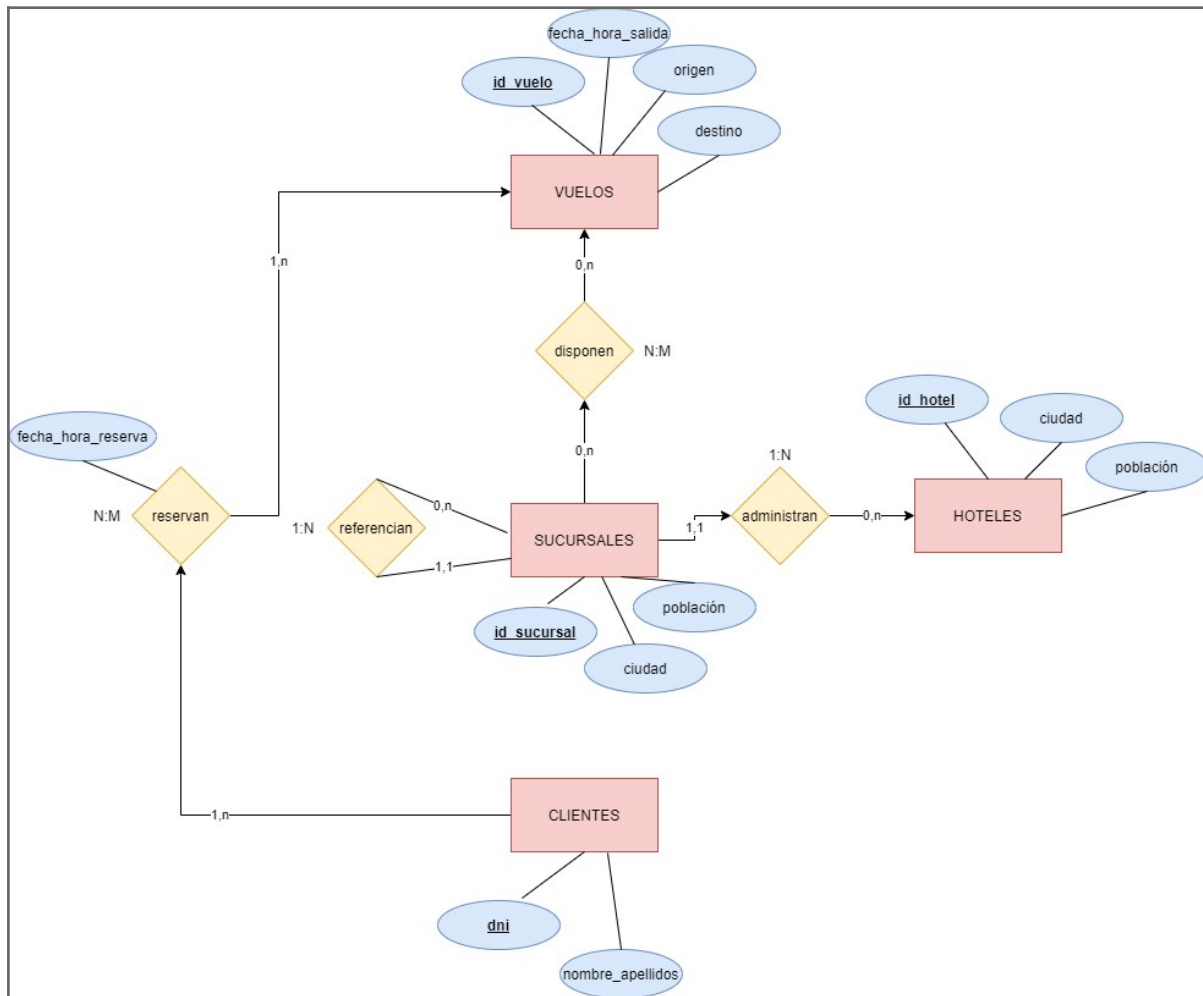


EJERCICIO DDL [4 puntos]

DEFINICIÓN



Crea las tablas correspondientes para el diagrama indicado arriba teniendo en cuenta las restricciones que se indican a continuación:

MODELO RELACIONAL

CLIENTES (dni, nombre_apellidos)

RESERVAN (dni_clientes, id_vuelo, fecha_hora_reserva)

VUELOS (id_vuelo, fecha_hora_salida, origen, destino)

DISPONEN (id_vuelo, id_sucursal)

SUCURSALES (id_sucursal, ciudad_sucursal, poblacion_sucursales, id_sucursal_ref)

HOTELES (id_hotel, ciudad_hoteles, poblacion_hoteles, id_sucursal)

- 1 Todas las claves foráneas deben actualizarse y borrarse cuando se actualicen y borren las claves de las tablas padre.
- 2 Queremos poder referenciar todas las restricciones de claves foráneas, usa nombres que identifiquen de forma correcta cada restricción
- 3 El atributo "fecha_hora_reserva" se corresponde con la fecha y hora en que el cliente reserva el vuelo. No queremos restringir la reserva por cliente a solo una, puesto que puede ocurrir que el mismo cliente reserve varias plazas para el mismo vuelo. Teniendo esto en cuenta, configura como creas conveniente este atributo en la tabla "reservan".
- 4 Toda fecha y hora debe tener por defecto la fecha y hora actual.
- 5 La mayoría de sucursales y hoteles se encuentran en la ciudad de Sevilla, pon este valor por defecto en los campos correspondientes.
- 6 Queremos definir todos los id como numéricos enteros de 4 dígitos
- 7 Debemos asegurarnos de que el valor que se introduce en el "dni" del cliente cumple con un patrón propio de un dni, teniendo en cuenta que solo vamos a introducir los dígitos de éste (NNNNNNNN).
- 8 Debes comprobar que las cláusulas que has añadido al script para cumplir los requisitos de los apartados 1, 4 y 7 funcionan. Explica aquí cómo realizas la comprobación (adjunta las capturas de pantalla que creas necesarias)

INSERCIÓN DE DATOS

- Debes insertar al menos dos tuplas de cada tabla. Copia aquí el script correspondiente

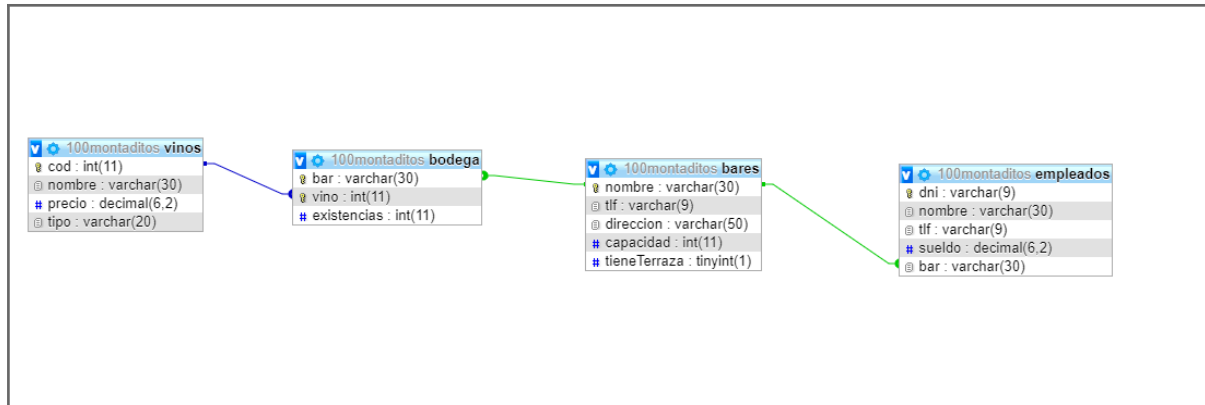
MODIFICACIONES

Una vez diseñada y creada nuestra base de datos, nos damos cuenta de que nos faltan algunos detalles por implementar:

- Queremos agregar el número de asiento que reserva cada cliente. Explica aquí dónde crees que es más lógico guardar esta información y añade al final del script la cláusula o cláusulas que aplicarían este cambio a la base de datos.
- Queremos cambiar la forma en que guardamos los nombre de los clientes, nos interesa que el nombre y los apellidos estén en dos columnas separadas. Añade al final del script la cláusula o cláusulas que aplicarían este cambio a la base de datos.

CONSULTAS [6 puntos]

Diagrama relacional de la base de datos:



A partir de la base de datos “100 montaditos”, realiza las consultas que se describen a continuación, teniendo en cuenta los siguientes criterios:

- Utiliza alias en los casos necesarios para que el resultado sea claro.
 - Muestra los datos en el orden en que se pide
 - Si la consulta no se ejecuta, no será evaluada
- 1 Mostrar todos los datos de los empleados que trabajan en bares que no tienen terrazas. **0,6 puntos**

```

SELECT empleados.*
FROM empleados
INNER JOIN bares
ON bares.nombre = empleados.bar
WHERE bares.tieneTerraza = 1;
  
```

- 2 Realizar una consulta que muestre todos los datos **del vino o los vinos** más baratos que existen. **0,6 puntos**

```

SELECT vinos.*
FROM vinos
WHERE precio <= ALL (
    SELECT precio
    FROM vinos
);
  
```

- 3 Mostrar el nombre de los bares que sirvan los vinos más caros junto con el código del vino y las existencias que tiene en este momento **0,9 puntos**

```
SELECT bares.nombre, vinos.cod, bodega.existencias
FROM bares
INNER JOIN bodega
ON bodega.bar = bares.nombre
INNER JOIN vinos
ON bodega.vino = vinos.cod
WHERE precio >= ALL (
    SELECT precio
    FROM vinos
);
```

- 4 Un bar de calidad es aquel que tiene un empleado por cada cliente. Averigua los bares que cumplen este requisito. La consulta debe mostrar: el nombre del bar, el número de empleados y la capacidad **1,2 puntos**

```
SELECT bares.nombre, COUNT(empleados.bar), bares.capacidad
FROM bares
INNER JOIN empleados
ON empleados.bar = bares.nombre
GROUP BY bares.nombre
HAVING COUNT(empleados.bar) >= bares.capacidad;
```

- 5 Queremos saber los bares que tienen empleados. Muestra los siguientes datos: nombre del bar, nombre del empleado y sueldo de éste. **0,6 puntos**

```
SELECT bares.nombre, empleados.nombre, empleados.sueldo
FROM bares
INNER JOIN empleados
ON bares.nombre = empleados.bar
WHERE bares.nombre = empleados.bar;
```

- 6 Mostrar el nombre, precio y tipo de vino de todos los vinos disponibles en el “Bar Manolo”. El resultado debe mostrarse ordenado de mayor a menor precio **0,6 puntos**

```
SELECT vinos.nombre, vinos.precio, vinos.tipo
FROM vinos
INNER JOIN bodega
ON vinos.cod = bodega.vino
INNER JOIN bares
ON bares.nombre = bodega.bar
WHERE bodega.bar = 'Bar Manolo'
ORDER BY vinos.precio DESC;
```

- 7 Los bares sin terraza deben cerrar debido a las nuevas restricciones a causa del COVID-19. Muestra un listado de los empleados de estos bares para poder avisarlos. La consulta debe mostrar el nombre del bar, seguido del nombre y el número de teléfono de cada empleado. **Debes usar subconsultas. 0,6 puntos**

```
SELECT bar, nombre, tlf
FROM empleados
WHERE bar = ANY (
    SELECT nombre
    FROM bares
    WHERE tieneTerraza = 0
);
```

- 8 El empleado “Daniel Dado” está considerado como el mejor cocinero. Nos interesa mostrar en su perfil de nuestra web los vinos que se pueden degustar en el bar donde trabaja. La consulta debe mostrar todos los datos de dichos vinos. **0,9 puntos**

```
SELECT vinos.*
FROM vinos
INNER JOIN bodega
ON bodega.vino = vinos.cod
INNER JOIN empleados
ON empleados.bar = bodega.bar
WHERE empleados.nombre = 'Daniel Dado';
```