**Claves únicas**: \_\_179888\_\_\_\_179524\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nombres:** \_\_\_\_\_\_José Luis Gutiérrez Espinosa \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Bruno Vitte San Juan\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ejercicio 1 –** Completa o contesta verdadero (V) o falso (F)

1. \_\_\_\_IaaS\_\_\_\_\_ son servidores sin sistema operativo que se rentan a los clientes.
2. La restricción de \_\_\_\_\_\_integridad\_\_\_\_\_\_\_ hace que la base de datos tenga datos coherentes.
3. \_\_\_PaaS\_\_\_servicio en la nube que rentan las aplicaciones. El proveedor de la nube es responsable de toda la arquitectura incluyendo las aplicaciones y debe prestar el servicio a sus clientes siguiendo acuerdos de servicio específicos.
4. \_\_Administrador\_\_Es el responsable por la administración de los recursos tanto del DBMS como de las bases de datos que tenga la organización.
5. (\_F\_\_) Los atributos de un vínculo M-N se trasladan a la entidad con N.
6. Los atributos \_\_\_\_\_\_\_\_\_\_\_\_\_débiles\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ no son parte de la tabla.
7. Los DBMS relacionales utilizan el \_\_\_\_structured query language\_\_\_\_\_\_ como un lenguaje estándar.
8. Un \_\_\_\_\_\_vínculo\_\_\_\_\_\_\_es una asociación entre entidades, normalmente, de tipos diferentes.
9. La restricción de \_\_\_\_\_\_\_\_\_\_\_\_participación\_\_\_\_\_\_\_\_\_\_\_ indica que la existencia de una entidad depende de estar relacionada con otra por medio de un tipo de relación.
10. Las entidades \_\_\_\_\_débiles\_\_\_\_\_\_\_\_ dependen de la existencia de otra entidad.

**Ejercicio 2 – Diagrama Entidad - Vínculo**

La Secretaría de Seguridad Pública de la CDMX desea realizar un sistema de información (diagrama entidad – vínculo) para llevar el registro de todas las personas que cometen un delito, así como la información referente a las organizaciones a las que pertenecen y los objetivos perseguidos por las mismas.

Se dispone de la siguiente información: las personas que cometen un delito llamados malhechores están definidos por un código, nombre, fecha de nacimiento, recompensa ofrecida y los distintos alias que puede tener. Los delitos por los que pueden ser acusados son: carteristas, falsificadores y estafadores. De los carteristas, se quiere almacenar además su país de origen y fecha de su primer delito. En cuanto a los falsificadores almacenaremos además su tipo y su especialidad. De los estafadores almacenaremos además una descripción adicional. Todos los malhechores, además, pueden ser reincidentes, en cuyo caso almacenaremos la fecha en la que cometieron el primer delito y el número total de delitos cometidos. Algunos carteristas se pueden considerar malhechores muy peligrosos, en cuyo caso se almacenará el número de víctimas que tienen. Cuando un agente es asignado a la captura de un malhechor se almacenará el código del agente, su nombre, dirección y teléfonos. Un agente puede tener asignados varios malhechores y cada uno de ellos sólo tendrá un agente asignado. En ningún caso existirán malhechores muy peligrosos sin agente asignado. Los malhechores pueden estar vinculados a organizaciones delictivas (Bandas mafiosas, terroristas, etc.) en cuyo caso se almacenará la fecha de ingreso y salida de los malhechores en las organizaciones. Un delincuente sólo pertenecerá a una organización en un momento dado, sin embargo debe quedar registro de todas las organizaciones a las que ha pertenecido. Finalmente, de las organizaciones se almacenará un código de organización, nombre, descripción, objetivo y fecha de creación.

**Ejercicio 3 – Modelo Relacional**

Transforma el siguiente diagrama entidad – vínculo a los esquemas correspondientes en el modelo relacional. Escribe los esquemas relacionales en el espacio provisto a continuación subrayando las claves primarias y señalando con las letras FK a las claves externas.



Sede(#sede, presupuesto)

Complejo(idC, localización, jefe, tipo, #sede(FK))

Unideportivo(idC)

Deporte(idC, deporte)

Polideportivo(idC, idA(FK))

Área(idA, localización, fecha)

Evento(idE, fecha, #comisarios, #participantes, duración, IdC(FK))

Participa(idE(FK), idCom(FK))

Comisario(idCom, tipo)

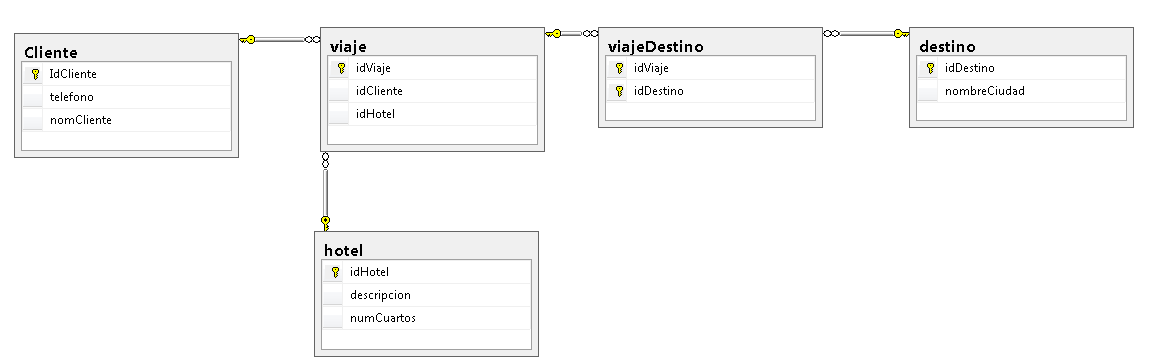
Juez(idCom, título, año)

Observador(idCom, tiempo)

**Ejercicio 4 – SQL**

A continuación se describen las características principales del sistema de bases de datos para el registro de clientes de hoteles en México.

La siguiente figura muestra el modelo conceptual del sistema de base de datos.



El archivo **CrearBD.txt** contienen las instrucciones SQL necesarias para definir la estructura de la base de datos del sistema de hoteles.

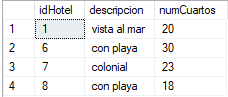
Ingresa al DBMS SQL Server, crea una nueva base de datos llamada **vacaciones**, ejecuta las sentencias SQL incluidas en el archivo mencionado,

Para poder responder las preguntas siguientes debes llenar la base de datos con el archivo llamado **InsertarBD** que determina el estado inicial de la base de datos.

Posteriormente responde las siguientes preguntas elaborando las consultas que correspondan. Después de resolver cada consulta, copia y pega la sentencia y la tabla de resultados después de cada pregunta.

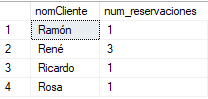
--1. [0.7] Listar toda la información de los hoteles con número de cuartos mayor a 8.

select \* from hotel where numCuartos > 8



-- 2. [0.7] Mostrar el nombre y la cantidad de viajes de cada cliente.

select nomCliente, count (\*) as num\_reservaciones from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente group by nomCliente



-- 3. [0.8] Listar el nombre de los clientes que visitan a dos ciudades como mínimo.

select distinct nomCliente from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente

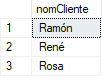
inner join viajeDestino on viaje.idViaje = viajeDestino.idViaje

inner join destino on viajeDestino.idDestino = destino.idDestino

where (select count (destino.idDestino) from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente

inner join viajeDestino on viaje.idViaje = viajeDestino.idViaje

inner join destino on viajeDestino.idDestino = destino.idDestino)> 1



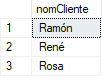
-- 4. [0.8] Listar el nombre y el teléfono de todos los clientes que tuvieron una cantidad de viajes superior a 2.

select distinct nomCliente from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente

inner join viajeDestino on viaje.idViaje = viajeDestino.idViaje

inner join destino on viajeDestino.idDestino = destino.idDestino

where (select count (idCliente) from Cliente)> 2



-- 5. [1.0] Mostrar el nombre de los clientes que estuvieron en hoteles con playa o que visitaron Oaxaca.

select distinct nomCliente from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente

inner join viajeDestino on viaje.idViaje = viajeDestino.idViaje

inner join destino on viajeDestino.idDestino = destino.idDestino

inner join hotel on hotel.idHotel = viaje.idHotel

where destino.nombreCiudad like 'Oaxaca' or hotel.descripcion like 'playa'



-- 6. [1.0] Mostrar el nombre de los clientes que visitaron Acapulco, pero no Cancún.

select distinct nomCliente from Cliente inner join viaje on Cliente.IdCliente = viaje.idCliente

inner join viajeDestino on viaje.idViaje = viajeDestino.idViaje

inner join destino on viajeDestino.idDestino = destino.idDestino

inner join hotel on hotel.idHotel = viaje.idHotel

where destino.nombreCiudad like 'Acapulco' and destino.nombreCiudad not like 'Cancun'

