

Universidad Argentina de la Empresa

Trabajo práctico: Modelo de datos crucero



Integrantes:

- Asplanatti Enzo Andrea - Lu: 1145364
- Berntsen Nikolas - Lu: 1154983
- Lane Gonzalo - Lu: 1150170
- Marcos Juan Cruz - Lu: 1148578
- Oliveira Emilio – Lu: 1150468

Profesores:

- Escandell Gustavo
- Maldonado Charles

Curso y asignatura:

- Ingeniería de datos I. Jueves - Mañana

-----2023-----

Universo del discurso / Historia de usuario:

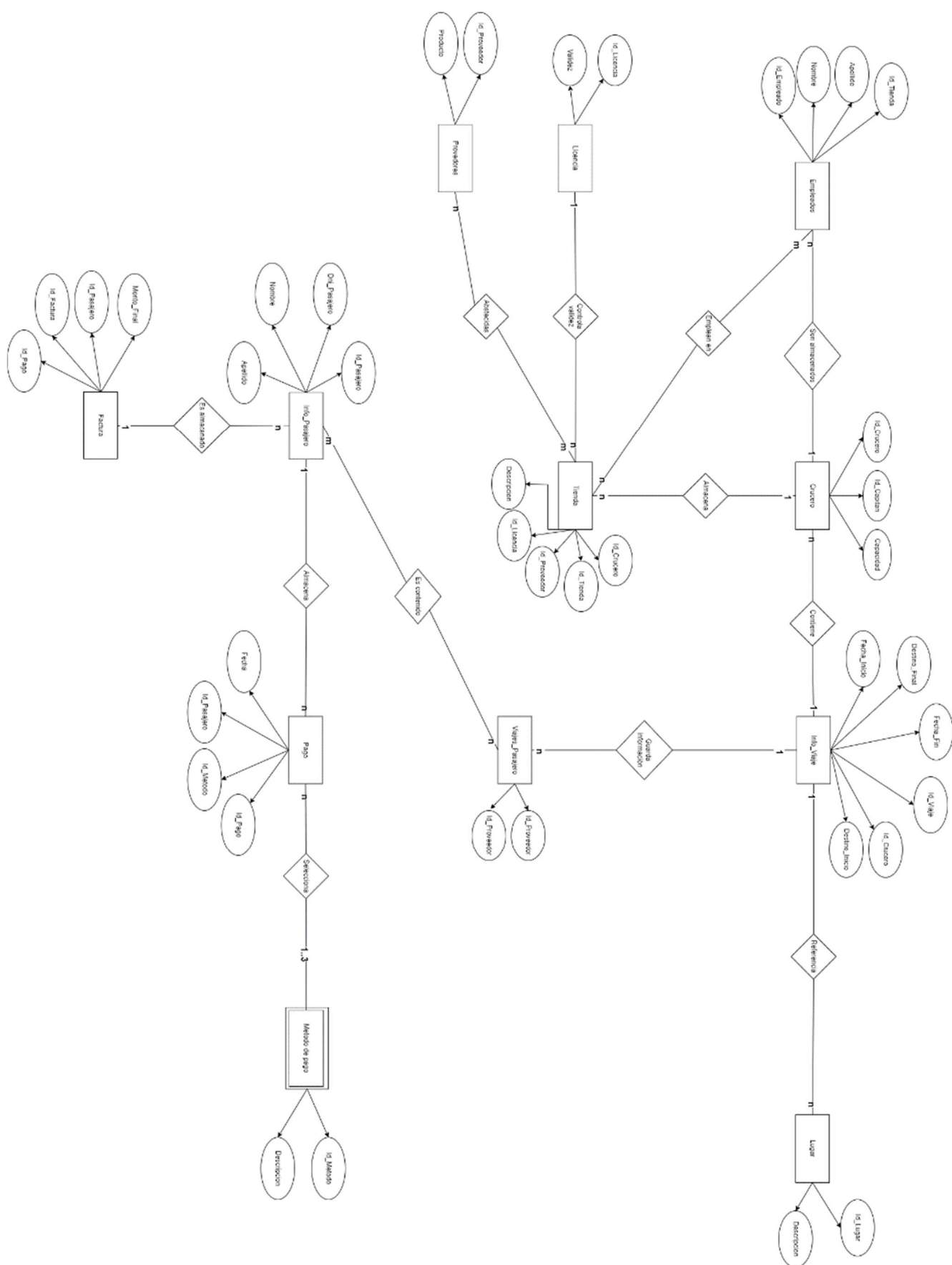
La empresa de cruceros busca una base de datos que incluya los registros entre los pasajeros, los empleados, las tiendas dentro de los viajes y consumiciones dentro de los mismos. El cliente debe tener registrado su documento único de identidad (DNI), su id de pasajero univoco, para poder identificar las acciones tomadas dentro del crucero. Estas acciones pueden ser tomadas en diferentes sectores dentro del crucero, como en diferentes locales. Cada pago que realice también será guardado e identificado, por quien fue, en que momento y en que sector dentro del crucero. Cada uno de ellos puede ser identificado, mediante un numero único de pago, junto con un id de producto, para saber lo que se pagó, junto con el medio de pago utilizado (efectivo o tarjeta).

También queremos registrar dentro de nuestro sistema, todos los ingresos y egresos de productos de las tiendas de cada crucero (esto se vera reflejado en ventas y al momento de ser provisionadas). Cada crucero debe tener contemplada la capacidad de pasajeros, su patente y si esta disponible o en viaje. También su cantidad de viajes, que estos van a estar identificados por su id de viaje y duración. En lo que los empleados pueden aparecer registrados en diferentes cruceros. Los empleados deben tener su id de empleado, como su hora de ingreso y egreso, para registrarlos y validar si esta disponible para una nueva embarcación.

Cada tienda va a tener su propia licencia, que se tomará como identificador de estos. Va a tener alguna identificación válida como un "0" para licencia inválida y un "1" si su id de licencia valida. Y los proveedores van a utilizar el numero vinculado de la tienda que le provee sus servicios. Luego cada proveedor va a tener su propio id, junto con el tipo de producto que distribuye.

Diagramas creados:

Diagrama de Entidad-Relación y diagrama físico



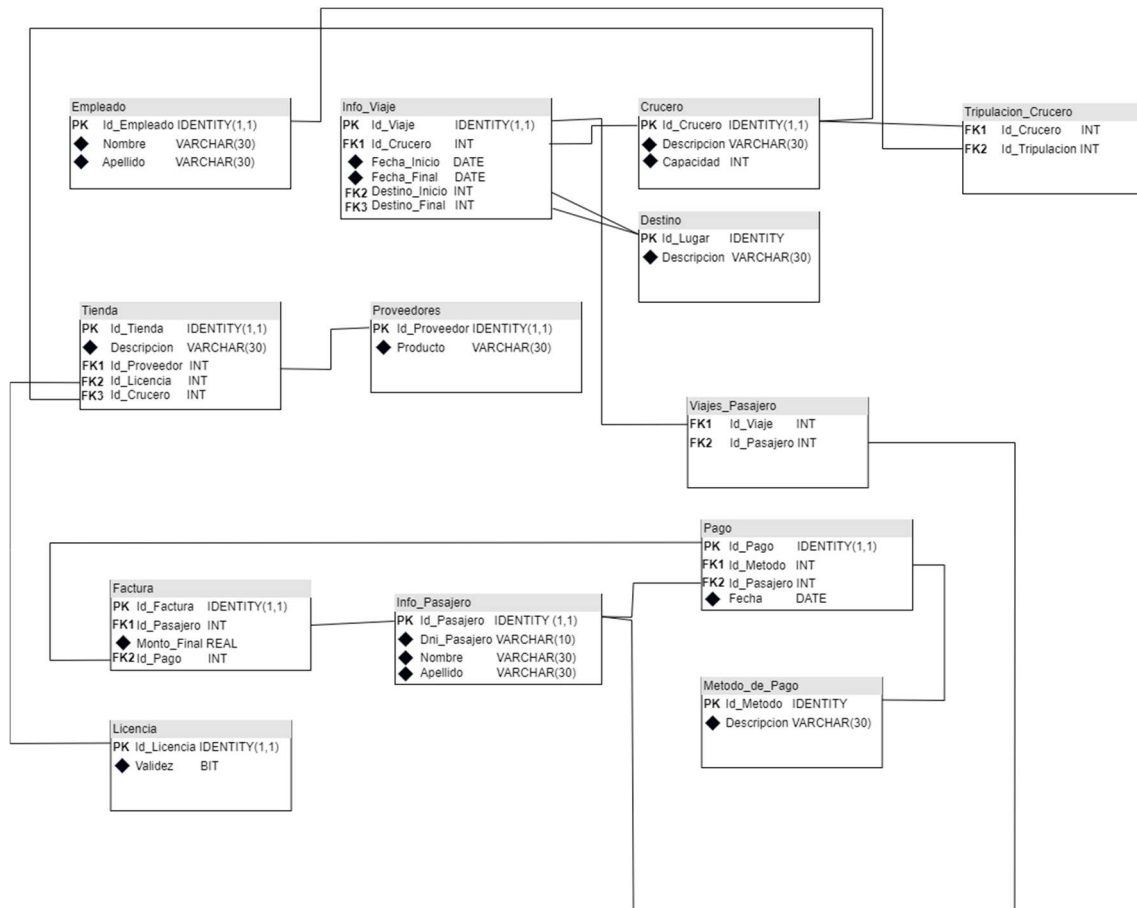
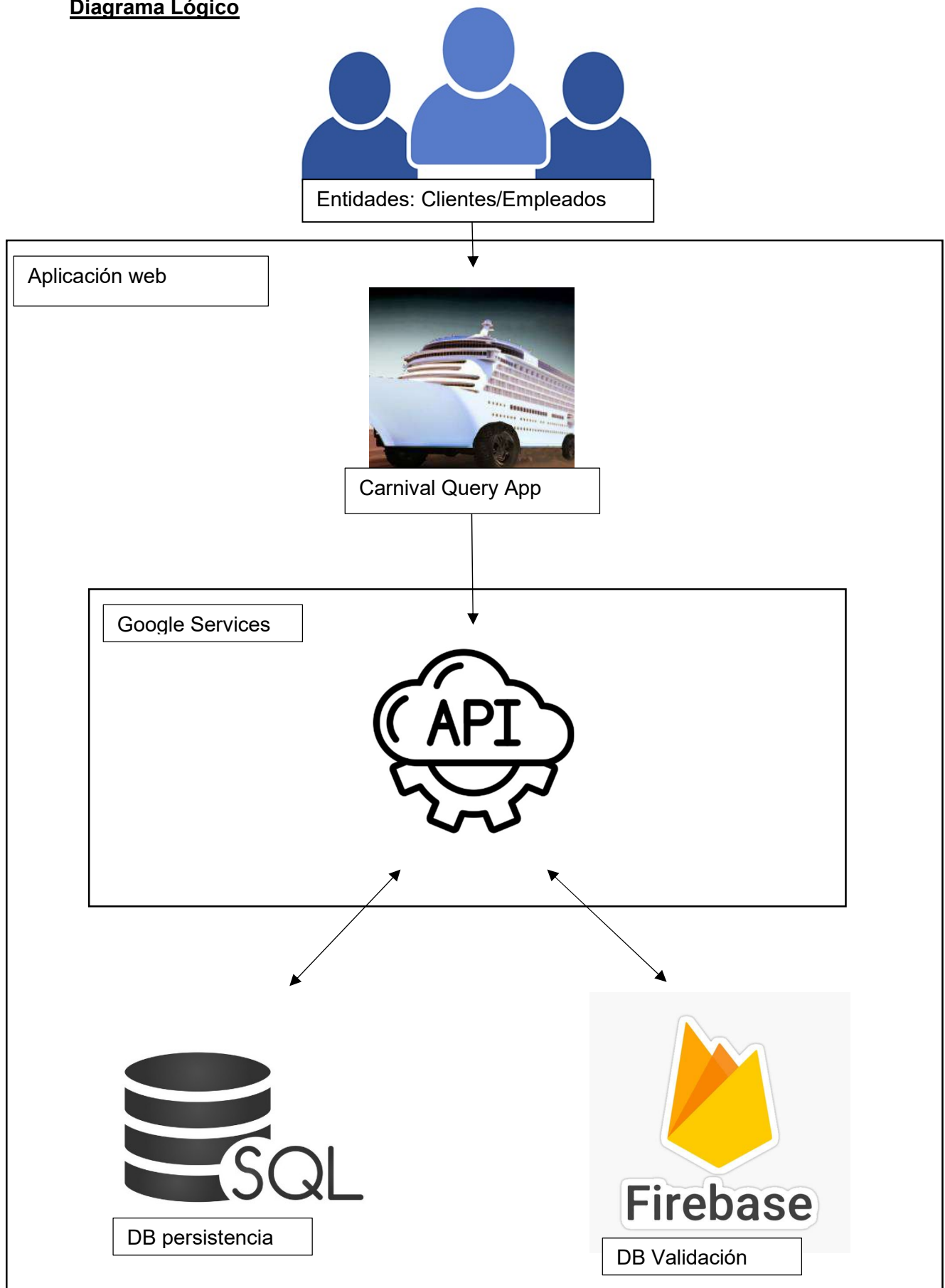


Diagrama Lógico



Diccionario de datos:

Tabla Empleado				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Empleado	int IDENTITY(1,1)	Numero de identificación de Empleado	
	Nombre	varchar(30)	Nombre del empleado	
	Apellido	varchar(30)	Apellido del emplado	

Tabla Info_Pasajero				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Pasajero	int IDENTITY(1,1)	Numero de identificación del Pasajero	
	DNI_Pasajero	varchar(10)	Documento Nacional de Identidad del Pasajero	
	Nombre	varchar(30)	Nombre del Pasajero	
	Apellido	varchar(30)	Apellido del Pasajero	

Tabla Proveedores				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Proveedor	int IDENTITY(1,1)	Numero de identificación de Proveedor	
	Producto	varchar(30)	Nombre de Producto	

Tabla Licencia				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Licencia	int IDENTITY(1,1)	Numero de identificación de Licencia	
	validez	bit	Valor de validación de Licencia	

Tabla Tienda				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Tienda	int IDENTITY(1,1)	Numero de identificación de la Tienda	
	Descripción	varchar(30)	Descripción de la Tienda	
FK	Id_Proveedor	int	Numero de identificación de Proveedor	
FK	Id_Licencia	int	Numero de identificación de Licencia	
FK	Id_Crucero	int	Numero de identificación del Crucero	

Tabla Crucero				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Crucero	int IDENTITY(1,1)	Numero de identificación del Crucero	
	Descripcion	varchar(30)	Nombre del Crucero	
	Capacidad	int	Capacidad total de pasajeros del curcero	

Tabla Metodo_De_Pago				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Metodo	int IDENTITY(1,1)	Numero de identificación del Metodo de Pago	
	Descripcion	varchar(30)	Tipo de Metodo de Pago	

Tabla Pago				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Pago	int IDENTITY(1,1)	Numero de identificación del Pago	
FK	Id_Metodo	int	Numero de identificación del Metodo de Pago	
FK	Id_Pasajero	int	Numero de identificación del Pasajero	
	Fecha	date	Fecha de realización del Pago	

Tabla Factura				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Factura	int IDENTITY(1,1)	Numero de identificación de Factura	
FK	Id_Pasajero	int	Numero de identificación del Pasajero	
	Monto_Final	real	Cantidad de Monto Final	
FK	Id_Pago	int	Numero de identificación del Pago	

Tabla Info_Viaje				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Viaje	int IDENTITY(1,1)	Numero de identificación del Viaje	
FK	Id_Crucero	int	Numero de identificación del Crucero	
	Fecha_Inicio	date	Fecha inicio del Viaje	
	Fecha_Fin	date	Fecha final del Viaje	
	Destino_Inicio	int	Destino inicial del Viaje	
	Destino_Fin	int	Destino final del Viaje	

Tabla Viajes_Pasajero				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Viaje	int	Numero de identificación del Viaje	
FK	Id_Pasajero	int	Numero de identificación del Pasajero	

Tabla Lugar				
	Columna	Tipo de Dato	Descripción	Null
PK	Id_Lugar	int IDENTITY(1,1)	Numero de identificación de Destino	
	Descripcion	varchar(30)	Nombre de Lugar de Destino	

Tabla Tripulacion_Crucereo				
	Columna	Tipo de Dato	Descripción	Null
FK	Id_Crucero	int	Numero de identificación del Crucero	
FK	Id_Tripulacion	int	Numero de identificación del Empleado	

Tablas

```
CREATE TABLE Info_Pasajero (  
    Id_Pasajero int IDENTITY(1,1) NOT NULL,  
    DNI_Pasajero varchar(10) NOT NULL,  
    Nombre varchar(30) NOT NULL,  
    Apellido varchar(30) NOT NULL,  
    CONSTRAINT PK_Pasajeros PRIMARY KEY (Id_Pasajero)  
);  
  
CREATE TABLE Proveedores (  
    Id_Proveedor int IDENTITY(1,1) NOT NULL,  
    Producto varchar(30),  
    CONSTRAINT PK_Proveedores PRIMARY KEY (Id_Proveedor)  
);  
  
CREATE TABLE Licencia (  
    Id_Licencia int IDENTITY(1,1) NOT NULL,  
    Validez bit NOT NULL,  
    CONSTRAINT PK_Licencia PRIMARY KEY (Id_Licencia)  
);  
  
CREATE TABLE Crucero (  
    Id_Crucero int IDENTITY(1,1) NOT NULL,  
    Descripcion varchar(30) NOT NULL,  
    Capacidad int NOT NULL,  
    CONSTRAINT PK_Crucero PRIMARY KEY (Id_Crucero)  
);  
  
CREATE TABLE Tienda (  
    Id_Tienda int IDENTITY(1,1) NOT NULL,  
    Descripcion varchar(30) NOT NULL,  
    Id_Proveedor int NOT NULL,  
    Id_Licencia int NOT NULL,  
    Id_Crucero int NOT NULL,  
    CONSTRAINT PK_Tienda PRIMARY KEY (Id_Tienda),  
    CONSTRAINT FK_Tienda_Proveedor FOREIGN KEY (Id_Proveedor)  
        REFERENCES Proveedores(Id_Proveedor),  
    CONSTRAINT FK_Tienda_Licencia FOREIGN KEY (Id_Licencia)  
        REFERENCES Licencia(Id_Licencia),  
    CONSTRAINT FK_Tienda_Crucero FOREIGN KEY (Id_Crucero)  
        REFERENCES Crucero(Id_Crucero)  
);  
  
CREATE TABLE Empleado (  
    Id_Empleado int IDENTITY(1,1) NOT NULL,  
    Nombre varchar(30) NOT NULL,  
    Apellido varchar(30) NOT NULL,  
    Id_Tienda int NOT NULL,  
    CONSTRAINT PK_Empleados PRIMARY KEY (Id_Empleado),  
    CONSTRAINT FK_Empleados_Tienda FOREIGN KEY (Id_Tienda)  
        REFERENCES Tienda(Id_Tienda)  
);  
  
CREATE TABLE Metodo_De_Pago (  
    Id_Metodo int IDENTITY(1,1) NOT NULL,  
    Descripcion varchar(30) NOT NULL,  
    CONSTRAINT PK_Metodo_De_Pago PRIMARY KEY (Id_Metodo)
```



```

);

CREATE TABLE Pago (
    Id_Pago int IDENTITY(1,1) NOT NULL,
    Id_Metodo int NOT NULL,
    Id_Pasajero int NOT NULL,
    Fecha DATE NOT NULL,
    CONSTRAINT PK_Pago PRIMARY KEY (Id_Pago),
    CONSTRAINT FK_Pago_Metodo FOREIGN KEY (Id_Metodo)
        REFERENCES Metodo_De_Pago(Id_Metodo),
    CONSTRAINT FK_Pago_Info_Pasajero FOREIGN KEY (Id_Pasajero)
        REFERENCES Info_Pasajero(Id_Pasajero)
);

CREATE TABLE Factura (
    Id_Factura int IDENTITY(1,1) NOT NULL,
    Id_Pasajero int NOT NULL,
    Monto_Final real NOT NULL,
    Id_Pago int NOT NULL,
    CONSTRAINT PK_Factura PRIMARY KEY (Id_Factura),
    CONSTRAINT FK_Factura_Pasajero FOREIGN KEY (Id_Pasajero)
        REFERENCES Info_Pasajero(Id_Pasajero),
    CONSTRAINT FK_Factura_Pago FOREIGN KEY (Id_Pago)
        REFERENCES Pago(Id_Pago)
);

CREATE TABLE Destino(
    Id_Lugar int IDENTITY(1,1) NOT NULL,
    Descripcion VARCHAR(30),
    CONSTRAINT PK_Id_Lugar PRIMARY KEY (Id_Lugar)
);

CREATE TABLE Info_Viaje (
    Id_Viaje int IDENTITY(1,1) NOT NULL,
    Id_Crucero int NOT NULL,
    Fecha_Inicio DATE NOT NULL,
    Fecha_Fin DATE NOT NULL,
    Destino_Inicio int NOT NULL,
    Destino_Fin int NOT NULL,
    CONSTRAINT PK_Viaje PRIMARY KEY (Id_Viaje),
    CONSTRAINT FK_Viaje_Crucero FOREIGN KEY (Id_Crucero)
        REFERENCES Crucero(Id_Crucero),
    CONSTRAINT FK_Viaje_Destino_Inicio FOREIGN KEY (Destino_Inicio)
        REFERENCES Destino(Id_Lugar),
    CONSTRAINT FK_Viaje_Destino_Final FOREIGN KEY (Destino_Fin)
        REFERENCES Destino(Id_Lugar)
);

CREATE TABLE Viajes_Pasajero (
    Id_Viaje int not null,
    Id_Pasajero int not null,
    CONSTRAINT FK_Viajes_Pasajero FOREIGN KEY (Id_Viaje)
        REFERENCES Info_Viaje(Id_Viaje),
    CONSTRAINT FK_Viajes_Pasajero2 FOREIGN KEY (Id_Pasajero)
        REFERENCES Info_Pasajero(Id_Pasajero)
);

```

```

CREATE TABLE Tripulacion_Crucero (
    Id_Crucero INT not null,
    Id_Tripulacion INT not null,
    CONSTRAINT FK_Tripulacion_Crucero_Crucero FOREIGN KEY (Id_Crucero)
        REFERENCES Crucero(Id_Crucero),
    CONSTRAINT FK_Tripulacion_Crucero_Empleado FOREIGN KEY (Id_Tripulacion)
        REFERENCES Empleado(Id_Empleado)
);

```

	Id_Crucero	Descripcion	Capacidad
1	1	213	1
2	2	8545	2
3	3	6031	3
4	4	9378	4
5	5	8152	5
6	6	6689	6
7	7	7427	7
8	8	5243	8

	Id_Lugar	Descripcion
1	1	Caribe
2	2	Mediterráneo
3	3	Hawái
4	4	Alaska
5	5	Nueva Zelanda
6	6	Seychelles
7	7	Fiji
8	8	Islas Galápagos

	Id_Empleado	Nombre	Apellido	Id_Tienda
1	1	Juan	Pérez	1
2	2	María	Gómez	3
3	3	Carlos	López	4
4	4	Ana	Martí...	2
5	5	Pedro	Rodrí...	1
6	6	Laura	Femá...	8
7	7	Sergio	Gonz...	5
8	8	Camen	Díaz	2

	Id_Factura	Id_Pasajero	Monto_Final	Id_Pago
1	1	1	500	10
2	2	2	750	4
3	3	3	600	15
4	4	4	800	3
5	5	5	450	12
6	6	6	550	8
7	7	7	700	19
8	8	8	900	1

	Id_Pasajero	DNI_Pasajero	Nombre	Apellido
1	1	1234567890	Juan	Pérez
2	2	9876543210	María	Gómez
3	3	5678901234	Carlos	López
4	4	3456789012	Ana	Martí...
5	5	6789012345	Pedro	Rodrí...
6	6	4567890123	Laura	Femá...
7	7	2345678901	Sergio	Gonz...
8	8	8901234567	Camen	Díaz

	Id_Viaje	Id_Crucero	Fecha_Inicio	Fecha_Fin	Destino_Inicio	Destino_Fin
1	1	1	2023-12-01	2023-12-10	2	3
2	2	2	2023-12-05	2023-12-15	4	5
3	3	3	2023-12-10	2023-12-20	6	7
4	4	4	2023-12-15	2023-12-25	8	9
5	5	5	2023-12-20	2023-12-30	10	11
6	6	6	2023-12-25	2024-01-04	12	13
7	7	7	2024-01-01	2024-01-10	14	15
8	8	8	2024-01-05	2024-01-15	16	17

	Id_Licencia	Validez
1	1	1
2	2	0
3	3	1
4	4	0
5	5	1
6	6	0
7	7	1
8	8	0

	Id_Metodo	Descripcion
1	1	Tarjeta de Crédito
2	2	Tarjeta de Débito
3	3	Efectivo
4	4	Transferencia B...
5	5	PayPal

	Id_Pago	Id_Metodo	Id_Pasajero	Fecha
1	1	1	1	2023-11-01
2	2	2	2	2023-11-02
3	3	3	3	2023-11-03
4	4	4	4	2023-11-04
5	5	5	5	2023-11-05
6	6	1	6	2023-11-06
-	-	-	-	-----

	Id_Proveedor	Producto
1	1	Ropa
2	2	Bebidas Alcohólicas
3	3	Bebidas sin Alcohol
4	4	Comida
5	5	Zapatillas
6	6	Bazar
7	7	Cosméticos

	Id_Tienda	Descripcion	Id_Proveedor	Id_Licencia	Id_Crucero
1	1	Tienda de Bebidas sin Alcohol	3	2	1
2	2	Tienda de Bebidas sin Alcohol	3	2	2
3	3	Tienda de Bebidas sin Alcohol	3	2	3
4	4	Tienda de Comida	4	3	4
5	5	Tienda de Comida	4	1	5
6	6	Tienda de Comida	4	5	6
7	7	Tienda de Zapatillas	5	5	7
8	8	Tienda de Zapatillas	5	4	8

	Id_Crucero	Id_Tripulacion
1	1	1
2	2	6
3	3	12
4	4	19
5	5	20
6	6	15
7	7	13
8	8	7

	Id_Viaje	Id_Pasajero
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8

Consultas

--1 TODAS LAS TIENDAS CON LICENCIA VALIDA

```

select Tienda.Id_tienda, Tienda.Descripcion, Licencia.Validez
from Tienda
inner join Licencia
on Tienda.Id_Licencia = Licencia.Id_Licencia
where Licencia.Validez = 'True'

```

	Id_tienda	Descripcion	Validez
1	4	Tienda de Comida	1
2	5	Tienda de Comida	1
3	6	Tienda de Comida	1
4	7	Tienda de Zapatillas	1
5	9	Tienda de Zapatillas	1
6	10	Tienda de Bazar	1
7	11	Tienda de Bazar	1
8	13	Tienda de Cosméticos	1

--2 CANTIDAD DE EMPLEADOS QUE TIENE EL CRUCERO

```
select Crucero.Id_Crucero, count(Empleado.Id_Empleado) as cantidad_de_empleados
from Crucero
inner join Tripulacion_Crucero
on Crucero.Id_Crucero = Tripulacion_Crucero.Id_Crucero
inner join Empleado
on Empleado.Id_Empleado = Tripulacion_Crucero.Id_Tripulacion
group by Crucero.Id_Crucero
```

	Id_Crucero	cantidad_de_empleados
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1

--3 CUAL METODO DE PAGO SE USO EN CADA FACTURA

```
select Metodo_De_Pago.Descripcion, Factura.Id_Factura
from Metodo_De_Pago
inner join pago
on Metodo_De_Pago.Id_Metodo = Pago.Id_Metodo
inner join Factura
on Pago.Id_Pago = Factura.Id_Pago
```

	Descripcion	Id_Factura
1	PayPal	1
2	Transferencia Bancar	2
3	PayPal	3
4	Efectivo	4
5	Tarjeta de Débito	5
6	Efectivo	6
7	Transferencia Bancar	7
8	Tarjeta de Crédito	8

--4 A QUE CRUCERO PROVEE CADA PROVEEDOR

```
select crucero.Id_Crucero, Proveedores.Id_Proveedor
from Crucero
inner join tienda
on Crucero.Id_crucero = Tienda.Id_crucero
inner join Proveedores
on Tienda.Id_Crucero = Proveedores.Id_Proveedor
```

	Id_Crucero	Id_Proveedor
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7

--5 DESTINO DE CADA CRUCERO

```
select Crucero.Id_Crucero, Destino.Descripcion
from Crucero
inner join info_viaje
on Crucero.Id_Crucero = info_Viaje.Id_Crucero
inner join Destino
on Destino.Id_Lugar = info_Viaje.Destino_Fin
```

	Id_Crucero	Descripcion
1	1	Hawái
2	2	Nueva Zelanda
3	3	Fiyi
4	4	Bora Bora
5	5	Maldivas
6	6	Islas Griegas
7	7	Noruega
8	8	Tucuman

--6 TOSDOS LOS DESTINOS INICIALES Y FINALES

```
SELECT DISTINCT d1.Descripcion, d2.Descripcion FROM Info_Viaje
```

```

inner join Destino as d1
on d1.Id_Lugar = info_Viaje.Destino_Fin
inner join Destino as d2
on d2.Id_Lugar = info_Viaje.Destino_Inicio

```

	Descripcion	Descripcion
1	Bora Bora	Islas Galápagos
2	Canadá	Sudáfrica
3	Caribe	Australia
4	Fiji	Seychelles
5	Hawái	Mediterráneo
6	Islas Grieg...	Bahamas
7	Maldivas	Tahití
8	Noruega	Costa Rica

--7 MUETRA LA CANTIDAD DE VIAJER QUE HIZO CADA CRUCERO EN EL AÑO 2024

```

select Id_Crucero, count (id_viaje) as cantidad_de_viajes from Info_viaje
where year(Fecha_Inicio) = 2024
group by Id_Crucero

```

	Id_Crucero	cantidad_de_viajes
1	7	1
2	8	1
3	9	1
4	10	1
5	11	1
6	12	1
7	13	1
8	14	1

--8 CUAL ES EL METODO MAS USADO EN EL CRUCERO

```

select TOP 1 Metodo_De_Pago.Descripcion, COUNT(pago.Id_Pago) as metodo_de_pago
from pago
inner join Metodo_De_Pago
on Pago.id_metodo = metodo_de_pago.id_metodo
group by Metodo_De_Pago.Descripcion

```

	Descripcion	metodo_de_pago
1	Efectivo	4

--9 DATOS DE TODOS LOS VIAJES DE UN PASEJERO

```

select Info_Pasajero.Nombre, Info_Pasajero.Apellido, info_viaje.Id_Viaje,
d1.Descripcion, d1.Descripcion from Info_Pasajero
inner join Viajes_Pasajero
on Info_Pasajero.Id_Pasajero = Viajes_Pasajero.Id_Pasajero
inner join Info_Viaje
on Viajes_Pasajero.Id_Viaje = Info_Viaje.Id_Viaje
inner join Destino as d1
on d1.Id_Lugar = info_Viaje.Destino_Fin
inner join Destino as d2
on d2.Id_Lugar = info_Viaje.Destino_Inicio
where Info_Pasajero.Id_Pasajero = 12

```

	Nombre	Apellido	Id_Viaje	Descripcion	Descripcion
1	Isabel	Torres	12	Nueva Zelanda	Nueva Zelanda

--10 DESTINO MAS UTILIZADO

```
select top 1 d1.Descripcion, count(Viajes_Pasajero.Id_Viaje) as
cantidad_de_viajes from Info_Viaje
inner join Viajes_Pasajero
on Info_Viaje.Id_Viaje = Viajes_Pasajero.Id_Viaje
inner join Destino as d1
on d1.Id_Lugar = info_Viaje.Destino_Fin
group by d1.Descripcion
```

	Descripcion	cantidad_de_viajes
1	Bora Bora	2

--11 NOMBRE DE EMPLEADOS DE CADA CRUCERO

```
select Crucero.Id_Crucero, Empleado.Nombre, Empleado.Apellido from
Tripulacion_Crucero
inner join Crucero
on Tripulacion_Crucero.Id_Crucero = Crucero.Id_Crucero
inner join Empleado
on Tripulacion_Crucero.Id_Tripulacion = Empleado.Id_Empleado
```

	Id_Crucero	Nombre	Apellido
1	1	Juan	Pérez
2	2	Laura	Fernández
3	3	Isabel	Torres
4	4	Andrés	Santos
5	5	Eva	Jiménez
6	6	Pablo	Vargas
7	7	Miguel	Luna
8	8	Sergio	González

--12 NOMBRE DE LOS EMPLEADOS EN BARCOS CON TIENDAS NO VALIDAS

```
select Empleado.Nombre, Empleado.Apellido
from Licencia
inner join Tienda
on Tienda.Id_Licencia = Licencia.Id_Licencia
inner join Crucero
on Tienda.Id_Tienda = Crucero.Id_Crucero
inner join Tripulacion_Crucero
on Tripulacion_Crucero.Id_Crucero = Crucero.Id_Crucero
inner join Empleado
on Tripulacion_Crucero.Id_Tripulacion = Empleado.Id_Empleado
where Licencia.Validez = 'false'
```


	Nombre	Apellido
1	Juan	Pérez
2	Laura	Fernández
3	Isabel	Torres
4	Sergio	González
5	Rosa	Pérez
6	Maria	Gómez
7	Carmen	Díaz
8	Marta	Ruiz

--13 QUE PRODUCTO ENTREGA CADA PROVEEDOR

```
select Id_Proveedor, Producto from Proveedores
```

	Id_Proveedor	Producto
1	1	Ropa
2	2	Bebidas Alcohólicas
3	3	Bebidas sin Alcohol
4	4	Comida
5	5	Zapatillas
6	6	Bazar
7	7	Cosméticos

--14 MONTO GASTADO DE CADA PASAJERO

```
select info_pasajero.Id_Pasajero, sum(Factura.Monto_Final) as gasto_total from
Info_Pasajero
inner join Factura
on Info_Pasajero.Id_Pasajero = Factura.Id_Pasajero
group by info_pasajero.Id_Pasajero
order by sum(Factura.Monto_Final) asc
```

	Id_Pasajero	gasto_total
1	10	350
2	9	400
3	5	450
4	16	470
5	1	500
6	15	530
7	6	550
8	13	580

--15 CUANTOS VIAJES LLEGAN A TUCUMAN

```
select Id_Viaje from Info_Viaje
inner join Destino as d1
on d1.Id_Lugar = info_Viaje.Destino_Fin
where d1.descripcion= 'Tucuman'
```

	Id_Viaje
1	8
2	18

--16 PASAJEROS QUE EXEDIERON PAGOS DE 600

```
select info_pasajero.Id_Pasajero, sum(Factura.Monto_Final) as gasto_mayor_a_2000
from Info_Pasajero
inner join Factura
on Info_Pasajero.Id_Pasajero = Factura.Id_Pasajero
```

```
group by info_pasajero.Id_Pasajero
having sum(Factura.Monto_Final)>600
order by sum(Factura.Monto_Final) asc
```

	Id_Pasajero	gasto_mayor_a_2000
1	11	650
2	18	690
3	7	700
4	12	720
5	20	720
6	2	750
7	19	770
8	4	800

--17 CUANTOS PROVEEDORES TIENEN CADA TIENDA

```
select Tienda.Id_Tienda, count(Proveedores.Id_Proveedor) as
Cantidad_de_proveedores from Tienda
inner join Proveedores
on tienda.Id_Proveedor = Proveedores.Id_Proveedor
group by Tienda.Id_Tienda
```

	Id_Tienda	Cantidad_de_proveedores
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1

--18 CUANTA CAPACIDAD TIENE CADA CRUCERO

```
select Id_Crucero, Capacidad from Crucero
```

	Id_Crucero	Capacidad
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8

--19 QUE PAGO REALIZO CADA PASAJERO

```
select Metodo_De_Pago.Descripcion, Info_Pasajero.Id_Pasajero from pago
inner join Metodo_De_Pago
on pago.Id_Metodo = Metodo_De_Pago.Id_Metodo
inner join Info_Pasajero
on pago.Id_Pasajero = Info_Pasajero.Id_Pasajero
```

	Descripcion	Id_Pasajero
1	Tarjeta de Crédito	1
2	Tarjeta de Débito	2
3	Efectivo	3
4	Transferencia B...	4
5	PayPal	5
6	Tarjeta de Crédito	6
7	Tarjeta de Débito	7
8	Efectivo	8

--20 LOS DATOS DE TODOS LOS VIAJES

```
select Id_Viaje, Fecha_Inicio, Fecha_Fin, d2.Descripcion, d1.Descripcion from
Info_Viaje
inner join Destino as d1
on d1.Id_Lugar = info_Viaje.Destino_Fin
inner join Destino as d2
on d2.Id_Lugar = info_Viaje.Destino_Inicio
```

	Id_Viaje	Fecha_Inicio	Fecha_Fin	Descripcion	Descripcion
1	1	2023-12-01	2023-12-10	Mediterráneo	Hawái
2	2	2023-12-05	2023-12-15	Alaska	Nueva Zelanda
3	3	2023-12-10	2023-12-20	Seychelles	Fiya
4	4	2023-12-15	2023-12-25	Islas Galáp...	Bora Bora
5	5	2023-12-20	2023-12-30	Tahití	Maldivas
6	6	2023-12-25	2024-01-04	Bahamas	Islas Griegas
7	7	2024-01-01	2024-01-10	Costa Rica	Noruega
8	8	2024-01-05	2024-01-15	Japón	Tucuman

Procedimientos almacenados

-- FUNCION PARA AGREGAR DATOS A LA TABLA EMPLEADOS

```
create procedure Llenar_Empleado

    @Nombre varchar(30),
    @Apellido varchar(30),
    @Id_Tienda int

as
begin
    insert into Empleado ( Nombre, Apellido, Id_Tienda)
    values ( @Nombre, @Apellido, @Id_Tienda);
end;
go
```

-- FUNCION PARA AGREGAR DATOS A LA TABLA VIAJES

```
create procedure Llenar_Info_Viaje

    @Id_Crucero int,
    @Fecha_Inicio date,
    @Fecha_Fin date,
    @Destino_Inicio int,
```

```

        @Destino_fin int

as
begin
    insert into Info_Viaje (Id_Crucero, Fecha_Inicio, Fecha_Fin,
Destino_Inicio, Destino_fin)
        values (@Id_Crucero, @Fecha_Inicio, @Fecha_Fin, @Destino_Inicio,
@Destino_fin);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA CRUCERO

create procedure Llenar_Crucero

    @Capacidad int,
    @Descripcion varchar(30)

as
begin
    insert into Crucero (Capacidad, Descripcion)
        values (@Capacidad, @Descripcion);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA TIENDAS

create procedure Llenar_Tienda

    @Descripcion varchar(30),
    @Id_proveedor int,
    @Id_Licencia int,
    @Id_Crucero int

as
begin
    insert into Tienda ( Descripcion,Id_proveedor,Id_Crucero, Id_Licencia)
        values ( @Descripcion,@Id_proveedor, @Id_Licencia ,@Id_Crucero);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA PROVEEDORES

create procedure Llenar_Proveedores

    @Producto varchar(30)

as
begin
    insert into Proveedores ( Producto)
        values ( @Producto);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA VENTAS

create procedure Llenar_pago

    @Id_Metodo int,

```

```

        @Id_pasajero int,
        @Fecha date

as
begin
    insert into pago (Id_Metodo,Id_pasajero,Fecha)
    values (@Id_Metodo,@Id_pasajero, @Fecha);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA FACTURA

create procedure Llenar_Factura

        @Id_pasajero int,
        @Monto_Final int,
        @Id_Pago int

as
begin
    insert into Factura (Id_pasajero, Monto_Final, Id_Pago)
    values (@Id_pasajero, @Monto_Final, @Id_Pago);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA METODO DE PAGO

create procedure Llenar_Metodo_De_Pago

        @Descripcion varchar(20)

as
begin
    insert into Metodo_De_Pago ( Descripcion)
    values ( @Descripcion);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA METODO DE PAGO

create procedure Llenar_info_Pasajero

        @Dni_Pasajero varchar(10),
        @Nombre varchar(30),
        @apellido varchar(30)

as
begin
    insert into info_Pasajero ( Dni_Pasajero, Nombre, apellido)
    values ( @Dni_Pasajero, @Nombre, @apellido);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA LICENCIA

create procedure Llenar_Licencia

        @Validez bit

as

```

```

begin
    insert into Licencia ( Validez)
    values ( @Validez);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA DESTINO

create procedure Llenar_destino

    @Descripcion varchar (30)

as
begin
    insert into destino ( Descripcion)
    values ( @Descripcion);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA VIAJES_PASAJERO

create procedure Llenar_Viajes_Pasajero

    @Id_Viaje int,
    @Id_pasajero int

as
begin
    insert into Viajes_Pasajero (Id_Viaje, Id_pasajero)
    values (@Id_Viaje, @Id_pasajero);
end;
go

-- FUNCION PARA AGREGAR DATOS A LA TABLA TRIPULACION_CRUCERO

create procedure Llenar_Tripulacion_Crucero

    @Id_Crucero int,
    @Id_Tripulacion int

as
begin
    insert into Tripulacion_Crucero (Id_Crucero, Id_Tripulacion)
    values (@Id_Crucero, @Id_Tripulacion);
end;
go

```

“Exects” de los procedimientos

```

EXEC Llenar_info_Pasajero '1234567890', 'Juan', 'Pérez';
EXEC Llenar_info_Pasajero '9876543210', 'María', 'Gómez';
EXEC Llenar_info_Pasajero '5678901234', 'Carlos', 'López';
EXEC Llenar_info_Pasajero '3456789012', 'Ana', 'Martínez';
EXEC Llenar_info_Pasajero '6789012345', 'Pedro', 'Rodríguez';
EXEC Llenar_info_Pasajero '4567890123', 'Laura', 'Fernández';
EXEC Llenar_info_Pasajero '2345678901', 'Sergio', 'González';
EXEC Llenar_info_Pasajero '8901234567', 'Carmen', 'Díaz';
EXEC Llenar_info_Pasajero '1122334455', 'Javier', 'Sánchez';
EXEC Llenar_info_Pasajero '9988776655', 'Elena', 'Pardo';
EXEC Llenar_info_Pasajero '6677889900', 'Luis', 'Ramírez';
EXEC Llenar_info_Pasajero '1122112211', 'Isabel', 'Torres';

```

EXEC Llenar_info_Pasajero '3344556677', 'Miguel', 'Luna';
EXEC Llenar_info_Pasajero '5566778899', 'Lorena', 'Cabrerena';
EXEC Llenar_info_Pasajero '7799887766', 'Pablo', 'Vargas';
EXEC Llenar_info_Pasajero '1122993344', 'Rosa', 'Pérez';
EXEC Llenar_info_Pasajero '9988776655', 'Antonio', 'García';
EXEC Llenar_info_Pasajero '5566778899', 'Marta', 'Ruiz';
EXEC Llenar_info_Pasajero '3344556677', 'Andrés', 'Santos';
EXEC Llenar_info_Pasajero '7799887766', 'Eva', 'Jiménez';

EXEC Llenar_Proveedores 'Ropa';
EXEC Llenar_Proveedores 'Bebidas Alcohólicas';
EXEC Llenar_Proveedores 'Bebidas sin Alcohol';
EXEC Llenar_Proveedores 'Comida';
EXEC Llenar_Proveedores 'Zapatillas';
EXEC Llenar_Proveedores 'Bazar';
EXEC Llenar_Proveedores 'Cosméticos';

EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;
EXEC Llenar_Licencia 1;
EXEC Llenar_Licencia 0;

EXEC Llenar_Metodo_De_Pago 'Tarjeta de Crédito';
EXEC Llenar_Metodo_De_Pago 'Tarjeta de Débito';
EXEC Llenar_Metodo_De_Pago 'Efectivo';
EXEC Llenar_Metodo_De_Pago 'Transferencia Bancaria';
EXEC Llenar_Metodo_De_Pago 'PayPal';

EXEC Llenar_pago 1, 1, '2023-11-01';
EXEC Llenar_pago 2, 2, '2023-11-02';
EXEC Llenar_pago 3, 3, '2023-11-03';
EXEC Llenar_pago 4, 4, '2023-11-04';
EXEC Llenar_pago 5, 5, '2023-11-05';
EXEC Llenar_pago 1, 6, '2023-11-06';
EXEC Llenar_pago 2, 7, '2023-11-07';
EXEC Llenar_pago 3, 8, '2023-11-08';
EXEC Llenar_pago 4, 9, '2023-11-09';

EXEC Llenar_pago 5, 10, '2023-11-10';
EXEC Llenar_pago 1, 11, '2023-11-11';
EXEC Llenar_pago 2, 12, '2023-11-12';
EXEC Llenar_pago 3, 13, '2023-11-13';
EXEC Llenar_pago 4, 14, '2023-11-14';
EXEC Llenar_pago 5, 15, '2023-11-15';
EXEC Llenar_pago 1, 16, '2023-11-16';
EXEC Llenar_pago 2, 17, '2023-11-17';
EXEC Llenar_pago 3, 18, '2023-11-18';
EXEC Llenar_pago 4, 19, '2023-11-19';
EXEC Llenar_pago 5, 20, '2023-11-20';

EXEC Llenar_Factura 1, 500, 10;
EXEC Llenar_Factura 2, 750, 4;
EXEC Llenar_Factura 3, 600, 15;
EXEC Llenar_Factura 4, 800, 3;
EXEC Llenar_Factura 5, 450, 12;
EXEC Llenar_Factura 6, 550, 8;
EXEC Llenar_Factura 7, 700, 19;
EXEC Llenar_Factura 8, 900, 1;
EXEC Llenar_Factura 9, 400, 18;
EXEC Llenar_Factura 10, 350, 9;
EXEC Llenar_Factura 11, 650, 5;
EXEC Llenar_Factura 12, 720, 17;
EXEC Llenar_Factura 13, 580, 7;
EXEC Llenar_Factura 14, 960, 6;
EXEC Llenar_Factura 15, 530, 11;
EXEC Llenar_Factura 16, 470, 2;
EXEC Llenar_Factura 17, 820, 13;
EXEC Llenar_Factura 18, 690, 16;
EXEC Llenar_Factura 19, 770, 14;
EXEC Llenar_Factura 20, 720, 20;

EXEC Llenar_Crucero 1, 213;
EXEC Llenar_Crucero 2, 8545;
EXEC Llenar_Crucero 3, 6031;
EXEC Llenar_Crucero 4, 9378;
EXEC Llenar_Crucero 5, 8152;
EXEC Llenar_Crucero 6, 6689;
EXEC Llenar_Crucero 7, 7427;
EXEC Llenar_Crucero 8, 5243;
EXEC Llenar_Crucero 9, 10556;
EXEC Llenar_Crucero 10, 8923;
EXEC Llenar_Crucero 11, 7324;
EXEC Llenar_Crucero 12, 7876;
EXEC Llenar_Crucero 13, 6867;
EXEC Llenar_Crucero 14, 9979;
EXEC Llenar_Crucero 15, 6103;
EXEC Llenar_Crucero 16, 9098;
EXEC Llenar_Crucero 17, 8106;
EXEC Llenar_Crucero 18, 7350;
EXEC Llenar_Crucero 19, 6987;

EXEC Llenar_Crucero 20, 7632;

EXEC Llenar_Tienda 'Tienda de Bebidas sin Alcohol', 3,1 ,2;
EXEC Llenar_Tienda 'Tienda de Bebidas sin Alcohol', 3, 2, 2;
EXEC Llenar_Tienda 'Tienda de Bebidas sin Alcohol', 3, 3, 2;
EXEC Llenar_Tienda 'Tienda de Comida', 4, 4, 3;
EXEC Llenar_Tienda 'Tienda de Comida', 4, 5, 1;
EXEC Llenar_Tienda 'Tienda de Comida', 4, 6, 5;
EXEC Llenar_Tienda 'Tienda de Zapatillas', 5, 7, 5;
EXEC Llenar_Tienda 'Tienda de Zapatillas', 5, 8, 4;
EXEC Llenar_Tienda 'Tienda de Zapatillas', 5, 9, 1;
EXEC Llenar_Tienda 'Tienda de Bazar', 6, 10, 5;
EXEC Llenar_Tienda 'Tienda de Bazar', 6, 11, 3;
EXEC Llenar_Tienda 'Tienda de Bazar', 6, 12, 4;
EXEC Llenar_Tienda 'Tienda de Cosméticos', 7, 13, 7;
EXEC Llenar_Tienda 'Tienda de Cosméticos', 7, 14, 9;
EXEC Llenar_Tienda 'Tienda de Cosméticos', 7, 15, 13;
EXEC Llenar_Tienda 'Tienda de Ropa', 1, 16, 1;
EXEC Llenar_Tienda 'Tienda de Bebidas Alcohólicas', 2, 17, 20;
EXEC Llenar_Tienda 'Tienda de Bebidas sin Alcohol', 3, 18, 16;
EXEC Llenar_Tienda 'Tienda de Comida', 4, 19, 14;
EXEC Llenar_Tienda 'Tienda de Zapatillas', 5, 20, 14;

EXEC Llenar_destino 'Caribe';
EXEC Llenar_destino 'Mediterráneo';
EXEC Llenar_destino 'Hawái';
EXEC Llenar_destino 'Alaska';
EXEC Llenar_destino 'Nueva Zelanda';
EXEC Llenar_destino 'Seychelles';
EXEC Llenar_destino 'Fiyi';
EXEC Llenar_destino 'Islas Galápagos';
EXEC Llenar_destino 'Bora Bora';
EXEC Llenar_destino 'Tahití';
EXEC Llenar_destino 'Maldivas';
EXEC Llenar_destino 'Bahamas';
EXEC Llenar_destino 'Islas Griegas';
EXEC Llenar_destino 'Costa Rica';
EXEC Llenar_destino 'Noruega';
EXEC Llenar_destino 'Japón';
EXEC Llenar_destino 'Tucuman';
EXEC Llenar_destino 'Sudáfrica';
EXEC Llenar_destino 'Canadá';
EXEC Llenar_destino 'Australia';

EXEC Llenar_Info_Viaje 1, '2023-12-01', '2023-12-10', 2, 3;
EXEC Llenar_Info_Viaje 2, '2023-12-05', '2023-12-15', 4, 5;
EXEC Llenar_Info_Viaje 3, '2023-12-10', '2023-12-20', 6, 7;
EXEC Llenar_Info_Viaje 4, '2023-12-15', '2023-12-25', 8, 9;
EXEC Llenar_Info_Viaje 5, '2023-12-20', '2023-12-30', 10, 11;
EXEC Llenar_Info_Viaje 6, '2023-12-25', '2024-01-04', 12, 13;

EXEC Llenar_Info_Viaje 7, '2024-01-01', '2024-01-10', 14, 15;
 EXEC Llenar_Info_Viaje 8, '2024-01-05', '2024-01-15', 16, 17;
 EXEC Llenar_Info_Viaje 9, '2024-01-10', '2024-01-20', 18, 19;
 EXEC Llenar_Info_Viaje 10, '2024-01-15', '2024-01-25', 20, 1;
 EXEC Llenar_Info_Viaje 11, '2024-01-20', '2024-01-30', 2, 3;
 EXEC Llenar_Info_Viaje 12, '2024-01-25', '2024-02-04', 4, 5;
 EXEC Llenar_Info_Viaje 13, '2024-02-01', '2024-02-10', 6, 7;
 EXEC Llenar_Info_Viaje 14, '2024-02-05', '2024-02-15', 8, 9;
 EXEC Llenar_Info_Viaje 15, '2024-02-10', '2024-02-20', 10, 11;
 EXEC Llenar_Info_Viaje 16, '2024-02-15', '2024-02-25', 12, 13;
 EXEC Llenar_Info_Viaje 17, '2024-02-20', '2024-02-29', 14, 15;
 EXEC Llenar_Info_Viaje 18, '2024-02-25', '2024-03-05', 16, 17;
 EXEC Llenar_Info_Viaje 19, '2024-03-01', '2024-03-10', 18, 19;
 EXEC Llenar_Info_Viaje 20, '2024-03-05', '2024-03-15', 20, 1;

EXEC Llenar_Viajes_Pasajero 1, 1;
 EXEC Llenar_Viajes_Pasajero 2, 2;
 EXEC Llenar_Viajes_Pasajero 3, 3;
 EXEC Llenar_Viajes_Pasajero 4, 4;
 EXEC Llenar_Viajes_Pasajero 5, 5;
 EXEC Llenar_Viajes_Pasajero 6, 6;
 EXEC Llenar_Viajes_Pasajero 7, 7;
 EXEC Llenar_Viajes_Pasajero 8, 8;
 EXEC Llenar_Viajes_Pasajero 9, 9;
 EXEC Llenar_Viajes_Pasajero 10, 10;
 EXEC Llenar_Viajes_Pasajero 11, 11;
 EXEC Llenar_Viajes_Pasajero 12, 12;
 EXEC Llenar_Viajes_Pasajero 13, 13;
 EXEC Llenar_Viajes_Pasajero 14, 14;
 EXEC Llenar_Viajes_Pasajero 15, 15;
 EXEC Llenar_Viajes_Pasajero 16, 16;
 EXEC Llenar_Viajes_Pasajero 17, 17;
 EXEC Llenar_Viajes_Pasajero 18, 18;
 EXEC Llenar_Viajes_Pasajero 19, 19;
 EXEC Llenar_Viajes_Pasajero 20, 20;

EXEC Llenar_Empleado 'Juan', 'Pérez', 1;
 EXEC Llenar_Empleado 'María', 'Gómez', 3;
 EXEC Llenar_Empleado 'Carlos', 'López', 4;
 EXEC Llenar_Empleado 'Ana', 'Martínez', 2;
 EXEC Llenar_Empleado 'Pedro', 'Rodríguez', 1;
 EXEC Llenar_Empleado 'Laura', 'Fernández', 8;
 EXEC Llenar_Empleado 'Sergio', 'González', 5;
 EXEC Llenar_Empleado 'Carmen', 'Díaz', 2;
 EXEC Llenar_Empleado 'Javier', 'Sánchez', 10;
 EXEC Llenar_Empleado 'Elena', 'Pardo', 11;
 EXEC Llenar_Empleado 'Luis', 'Ramírez', 15;
 EXEC Llenar_Empleado 'Isabel', 'Torres', 19;
 EXEC Llenar_Empleado 'Miguel', 'Luna', 20;
 EXEC Llenar_Empleado 'Lorena', 'Cabrerera', 11;
 EXEC Llenar_Empleado 'Pablo', 'Vargas', 4;
 EXEC Llenar_Empleado 'Rosa', 'Pérez', 2;
 EXEC Llenar_Empleado 'Antonio', 'García', 9;

```

EXEC Llenar_Empleado 'Marta', 'Ruiz',7;
EXEC Llenar_Empleado 'Andrés', 'Santos',1;
EXEC Llenar_Empleado 'Eva', 'Jiménez',16;

EXEC Llenar_Tripulacion_Crucero 1, 1; -- Juan Pérez
EXEC Llenar_Tripulacion_Crucero 2, 6; -- Laura Fernández
EXEC Llenar_Tripulacion_Crucero 3, 12; -- Isabel Torres
EXEC Llenar_Tripulacion_Crucero 4, 19; -- Marta Ruiz
EXEC Llenar_Tripulacion_Crucero 5, 20; -- Andrés Santos
EXEC Llenar_Tripulacion_Crucero 6, 15; -- Luis Ramírez
EXEC Llenar_Tripulacion_Crucero 7, 13; -- Miguel Luna
EXEC Llenar_Tripulacion_Crucero 8, 7; -- Sergio González
EXEC Llenar_Tripulacion_Crucero 9, 3; -- Carlos López
EXEC Llenar_Tripulacion_Crucero 10, 9; -- Antonio García
EXEC Llenar_Tripulacion_Crucero 11, 10; -- Javier Sánchez
EXEC Llenar_Tripulacion_Crucero 12, 16; -- Eva Jiménez
EXEC Llenar_Tripulacion_Crucero 13, 14; -- Pablo Vargas
EXEC Llenar_Tripulacion_Crucero 14, 5; -- Pedro Rodríguez
EXEC Llenar_Tripulacion_Crucero 15, 11; -- Elena Pardo
EXEC Llenar_Tripulacion_Crucero 16, 17; -- Rosa Pérez
EXEC Llenar_Tripulacion_Crucero 17, 2; -- María Gómez
EXEC Llenar_Tripulacion_Crucero 18, 8; -- Carmen Díaz
EXEC Llenar_Tripulacion_Crucero 19, 18; -- Martín Cabrera
EXEC Llenar_Tripulacion_Crucero 20, 4; -- Ana Martínez

```

Normalización

Para llevar a cabo la normalización de la base de datos de cruceros, emplearemos un modelo y una base de datos que tenga fuertes relaciones, con las diferentes entidades que la componen. Esto, con los siguientes puntos:

1. Identificación de entidades: El primer paso en el proceso de normalización es identificar las entidades principales en la base de datos. En este caso, identificamos entidades como pasajero/cliente, viaje, pago, método de pago, datos del crucero, empleados, datos de viajes, tienda y proveedores de las mismas, como los destinos.

2. Definición de atributos: Para cada entidad, determinamos los atributos que describen sus características. Por ejemplo, el proveedor tiene su propio ID y el producto que provee.

3. Eliminación de redundancias: Revisamos los atributos de cada entidad y eliminamos cualquier redundancia. Por ejemplo, para no tener en dos tablas distintas los nombres de los clientes, interactuamos con las demás tablas, bajo su ID único. Otro ejemplo, es el hecho de guardar en una tabla distinta los datos de los viajes, pero solo con los id de los viajes, empleados y los destinos.

4. Identificación de dependencias funcionales: Analizamos las dependencias funcionales entre los atributos de cada entidad. Por ejemplo, en la entidad pago y método de pago, la identificación de la dependencia es de seleccionar al menos 1 método de pago para realizar el pago.

5. Normalización en niveles: para este caso, usamos la técnica Divide&Conquer (divide y conquista), para eliminar la redundancia entre las entidades, como también ofrecer diferentes caminos para relacionar las entidades.

6. Definición de claves primarias y foráneas: Para mantener la integridad de los datos, asignamos claves primarias a cada entidad y utilizamos claves foráneas para establecer relaciones entre entidades.

7. Refinamiento y optimización: Revisamos y refinamos la estructura de la base de datos para mejorar el rendimiento y la eficiencia. Esto puede incluir la adición de índices, la desnormalización selectiva para mejorar el rendimiento de consultas frecuentes, entre otras técnicas.

Testing

La realización de pruebas en la base de datos es un paso esencial en el desarrollo y mantenimiento de sistemas aeroportuarios, ya que asegura la integridad, seguridad y eficiencia de la información gestionada. Los objetivos fundamentales del testing en la base de datos aeroportuaria son los siguientes:

- **Validar la exactitud de los datos:** Verificar que la información almacenada en la base de datos sea precisa y esté actualizada, asegurando que refleje de manera precisa la realidad de la empresa de cruceros.
- **Garantizar la consistencia de los datos:** Evaluar que los datos relacionados entre sí tengan un sentido acorde entre las entidades. Por ejemplo, entre los clientes con sus viajes realizados, junto con los destinos visitados. Desde el punto de vista de la empresa, entre los trabajadores y las tiendas que ofrece cada crucero.
- **Evaluar el rendimiento:** Medir y optimizar la velocidad de acceso y respuesta de la base de datos para asegurar que los diferentes sistemas de la empresa de cruceros puedan operar de manera eficiente.

La estrategia de testing se lleva a cabo mediante los siguientes pasos:

- **Definición de casos de prueba:** Establecer escenarios de prueba que abarquen diferentes aspectos del sistema de cruceros, como el registro de viajes, clientes en los mismos, gestión y registro de pagos, evaluando los rendimientos de por ejemplo los métodos de pago mas utilizados. Otro ejemplo puede ser, los destinos más solicitados.
- **Ejecución de pruebas:** Realizar pruebas exhaustivas utilizando los casos de prueba definidos. Durante esta etapa, se simulan diversas situaciones para validar la correcta interacción de los datos en la base de datos.
- **Análisis de resultados:** Evaluar los resultados de las pruebas, identificar y documentar los problemas encontrados. Posteriormente, se priorizan las correcciones necesarias y se llevan a cabo las acciones correctivas pertinentes.

En resumen, la estrategia de testing en la base de datos de cruceros, busca asegurar la calidad y confiabilidad de la información almacenada, así como la eficiencia del sistema en su conjunto, mediante la definición, ejecución y análisis de casos de prueba.