**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**Escuela de Tecnología**

# **Carátula**

**Obligatorio 1**

**Base de datos 2**



**Diego Vaitcunas - 289389     Ignacio Pataro - 288045**

**Docente: Alberto Villar**

**Grupo: M2C**

**Carrera:**

**Analista Programador**

**Analista en tecnología de la información**

**00/06/2023**

**ÍNDICE**

[**Carátula 1**](#_heading=h.gjdgxs)

[**1) Restricciones de integridad de la letra, sobre el script de creación de tablas proporcionado (mediante ALTER de las estructuras dadas). 3**](#_heading=h.8idjnt60tgq)

[**1- Creación de todas las claves primarias. 3**](#_heading=h.pr046bkzcen6)

[**2- Otras restricciones relacionadas a la tabla cliente. 3**](#_heading=h.m6tczdohl7vk)

[**4- Otras restricciones relacionadas a la tabla Container. 4**](#_heading=h.szqf96pl3ht)

[**5- Otras restricciones relacionadas a la tabla Carga. 4**](#_heading=h.iaq99jdisohj)

[**Creación de índices que consideramos. 5**](#_heading=h.l5d642p70zvb)

[-Tabla Cliente 5](#_heading=h.w8sm8ddb1qg1)

[-Tabla Avion 5](#_heading=h.b8qgd76fttyf)

[-Tabla DContainer 6](#_heading=h.hctmq28ymeso)

[-Tabla Aeropuerto 6](#_heading=h.7mo1tv2b5or8)

[-Tabla Carga 6](#_heading=h.iavbot783ym5)

[**Datos de prueba para el obligatorio de base de datos. 6**](#_heading=h.es4a6tibjsvl)

[**Ingreso completo de datos de prueba válidos. 6**](#_heading=h.1fob9te)

[**1- Ejemplos válidos. 6**](#_heading=h.fduua0c57qev)

[-Inserción en tabla cliente. 6](#_heading=h.b4ekd4p0ffn3)

[-Inserción en tabla Avión 7](#_heading=h.pu3dh460d4aq)

[-Inserción en tabla DContainer 7](#_heading=h.66yortoehe7s)

[-Inserción en tabla Aeropuerto 7](#_heading=h.o3yo9yci9fmt)

[-Inserción en tabla Carga 7](#_heading=h.9bshrwd0bn96)

[**2- Ejemplos Inválidos. 7**](#_heading=h.3ftclh6qhwzf)

[-Inserción en tabla avión 8](#_heading=h.dtt0s5avcaq)

[-Inserción en tabla Container 8](#_heading=h.s3jffno0jiaj)

[-Inserción en tabla Aeropuerto 9](#_heading=h.k0p7nf85sgt0)

[-Inserción en tabla Carga 9](#_heading=h.vhjv9ci8hrvo)

[**Consultas SQL 9**](#_heading=h.ih52w5ig112n)

[a. Mostrar los datos de los clientes que cargaron más kilos este año que el promedio total de kilos cargados por todos los clientes el año pasado: 9](#_heading=h.16wusnpfdsx4)

[b. Del total de kilos cargados por cada avión, mostrar cuál fue el mayor valor, cuál fue el promedio y cuál fue el menor valor: 10](#_heading=h.1vshesmhmyhn)

[c. Para cada tipo de contenedor, mostrar sus datos, la cantidad de cargas en los que fue utilizado y el total de kilos cargados, si algún tipo de contenedor nunca fue utilizado, también deben mostrarse sus datos: 10](#_heading=h.7rk0k0n43w9i)

[d. Mostrar los datos de los clientes que utilizaron todos los aviones disponibles para sus cargas: 10](#_heading=h.hbnwrpo5xcrt)

[e. Mostrar el identificador de la carga, la fecha y los nombres de los aeropuertos de origen y destino para todas las cargas del año actual que utilizan aviones con una capacidad mayor a las 100 toneladas. 10](#_heading=h.cb3cuaxlya2x)

[f. Mostrar los datos del aeropuerto que recibió la mayor cantidad de kilos de los últimos 5 años. 11](#_heading=h.9gec8eclrk4g)

[**Utilizando T-SQL realizar los siguientes ejercicios: 11**](#_heading=h.1cjnxfo7hs1w)

[a. Escribir un procedimiento almacenado que reciba como parámetros un rango de fecha y retorne también por parámetros el identificador de avión que cargó más kilos en dicho rango de fechas y el nombre del cliente que cargó más kilos en dicho rango (si hay más de uno, mostrar el primero). 11](#_heading=h.3znysh7)

[b. Realizar un procedimiento almacenado que, dadas las 3 medidas de un contenedor (largo x ancho x alto) retorne en una tabla los datos de los contenedores que coinciden con dichas medidas, de no existir ninguno se debe retornar un mensaje. 12](#_heading=h.2et92p0)

[c. Hacer una función que reciba un código de aeropuerto y retorne la cantidad de kilos recibidos de carga cuando ese aeropuerto fue destino. 12](#_heading=h.tyjcwt)

[d. Hacer una función que, para un cliente dado, retorne la cantidad total de kilos transportados por dicho cliente a aeropuertos de diferente país. 12](#_heading=h.3dy6vkm)

[**6. Escribir los siguientes disparadores : 13**](#_heading=h.1t3h5sf)

[a. Realizar un disparador que lleve un mantenimiento de la cantidad de cargas acumuladas de un cliente, este disparador debe controlar tanto los ingresos de cargas como el borrado de cargas. 13](#_heading=h.4d34og8)

[b. Hacer un disparador que, ante la modificación de cualquier medida de un contenedor, lleve un registro detallado en la tabla AuditContainer (ver estructura de la tabla en el anexo del presente obligatorio). 13](#_heading=h.2s8eyo1)

[c. Realizar un disparador que cuando se registra una nueva carga se valide que el avión tiene capacidad suficiente para almacenarla, esta verificación debe tener en cuenta todas las cargas que se están haciendo en ese avión en la misma fecha. 14](#_heading=h.17dp8vu)

# **Restricciones de integridad de la letra, sobre el script de creación de tablas proporcionado (mediante ALTER de las estructuras dadas).**

# **1- Creación de todas las claves primarias.**

ALTER TABLE Cliente

ADD CONSTRAINT PK\_Cliente PRIMARY KEY (cliID);

ALTER TABLE Avion

ADD CONSTRAINT PK\_Avion PRIMARY KEY (avionID);

ALTER TABLE DContainer

ADD CONSTRAINT PK\_DContainer PRIMARY KEY (dContID);

ALTER TABLE Aeropuerto

ADD CONSTRAINT PK\_Aeropuerto PRIMARY KEY (codIATA);

ALTER TABLE Carga

ADD CONSTRAINT PK\_Carga PRIMARY KEY (idCarga);

# **2- Otras restricciones relacionadas a la tabla cliente.**

-Email que sea único

ALTER TABLE Cliente

ADD CONSTRAINT Unique\_Cliente\_Email UNIQUE (cliMail);

3- Otras restricciones relacionadas a la tabla Avion.

-Longitud máxima para avionMat.

ALTER TABLE Avion

ALTER COLUMN avionMAT varchar(20) not null;

-Capacidad máxima

ALTER TABLE Avion

ADD CONSTRAINT CHK\_Avion\_Capacidad Check (avionCapacidad <= 150);

# **4- Otras restricciones relacionadas a la tabla Container.**

-Restricciones de medidas

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Largo CHECK (dContLargo <= 2.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Ancho CHECK (dContAncho <= 3.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Alto CHECK (dContAlto <= 2.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Capacidad CHECK (dContCapacidad <= 7);

# **5- Otras restricciones relacionadas a la tabla Carga.**

-Asignación de claves foráneas.

ALTER TABLE Carga

ALTER COLUMN dContID char(6)

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAvion FOREIGN KEY (avionID) REFERENCES Avion(avionID);

ALTER TABLE Carga

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAvion FOREIGN KEY (avionID) REFERENCES Avion(avionID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaDContainer FOREIGN KEY (dContID) REFERENCES DContainer(dContID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaCliente FOREIGN KEY (cliID) REFERENCES Cliente(cliID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAeropuertoOrigen FOREIGN KEY (aeroOrigen) REFERENCES Aeropuerto(codIATA);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAeropuertoDestino FOREIGN KEY (aeroDestino) REFERENCES Aeropuerto(codIATA);

-Para que no se repitan los campos, avionID, dContID y cargaFch.

ALTER TABLE Carga

ADD CONSTRAINT UQ\_Carga\_avionID\_dContID\_cargaFch UNIQUE (avionID, dContID, cargaFch);

-- Controla los diferentes tipos y estados de carga

ALTER TABLE Carga

ADD CONSTRAINT CHK\_Carga\_Tipos CHECK (cargaStatus IN ('R', 'C', 'T', 'D', 'E'));

# **Creación de índices que consideramos.**

**Los índices los separamos por tabla haciendo énfasis en su clave primaria, en las claves foráneas y en algunos casos en algún atributo.**

# **-Tabla Cliente**

Índice en PK y Cantidad de cargas.

CREATE INDEX INDICE\_Cliente\_cliMail ON Cliente(cliMail);

CREATE INDEX INDICE\_Cliente\_cliCantCargas ON Cliente(cliCantCargas);

# **-Tabla Avion**

Índice en PK

CREATE INDEX INDICE\_Avion\_avionID ON Avion(avionID);

# **-Tabla DContainer**

Índice en PK.

CREATE INDEX INDICE\_DContainer\_dContID ON DContainer(dContID);

# **-Tabla Aeropuerto**

Índice en PK.

CREATE INDEX INDICE\_Aeropuerto\_codIATA ON Aeropuerto(codIATA);

# **-Tabla Carga**

Índice en claves foráneas y en PK, los otros índices los consideramos importantes por el hecho que es una empresa de gestión de cargas y sería útil acceder a estos datos con mayor facilidad.

CREATE INDEX INDICE\_Carga\_idCarga ON Carga(idCarga);

CREATE INDEX INDICE\_Carga\_avionID ON Carga(avionID);

CREATE INDEX INDICE\_Carga\_dContID ON Carga(dContID);

CREATE INDEX INDICE\_Carga\_cargaFch ON Carga(cargaFch);

CREATE INDEX INDICE\_Carga\_cliID ON Carga(cliID);

CREATE INDEX INDICE\_Carga\_aeroOrigen ON Carga(aeroOrigen);

CREATE INDEX INDICE\_Carga\_aeroDestino ON Carga(aeroDestino);

CREATE INDEX INDICE\_Carga\_cargaStatus ON Carga(cargaStatus);

# **Datos de prueba para el obligatorio de base de datos.**

# **Ingreso completo de datos de prueba válidos.**

# **1- Ejemplos válidos.**

### -Inserción en tabla cliente.

INSERT INTO Cliente (cliNom, cliMail, cliCantCargas) VALUES

('Diego Vaitcunas', 'DiegoAlvezV@hotmail.com', 2),

('Ignacio Pataro', 'IgnacioPataro2005@gmail.com', 0),

('Alberto Villar', 'AlbertoVillar1990@gmail.com', 5),

('Sofía García', 'sofia.garcia@hotmail.com', 4),

('Gabriel Fernández', 'gabriel.fernandez@hotmail.com', 2),

('Valentina Ruiz', 'valentina.ruiz@gmail.com', 1),

('Lucas Morales', 'lucas.morales@hotmail.com', 3),

('Carolina Méndez', 'carolina.mendez@gmail.com', 0);

GO

### -Inserción en tabla Avión

INSERT INTO Avion (avionID, avionMAT, avionMarca, avionModelo, avionCapacidad) VALUES

('P007', 'AB123', 'Boeing', '747', 110),

('P008', 'CD456', 'Pluna', 'A380', 140),

('P009', 'EF789', 'Emiratos Arabes', 'E190', 80),

('P010', 'UV123', 'Airbus', 'A380', 140),

('P011', 'WX456', 'Boeing', '777', 150),

('P012', 'YZ789', 'Embraer', 'E190', 80),

('P013', 'AB234', 'Cessna', 'Citation CJ4', 6),

('P014', 'CD567', 'Bombardier', 'Global 6000', 10);

GO

### -Inserción en tabla DContainer

INSERT INTO DContainer (dContID, dContLargo, dContAncho, dcontAlto, dcontCapacidad) VALUES

('DCUno', 2.3, 2.6, 2.3, 3),

('DCDos', 2.1, 3.4, 2.2, 4),

('DCTres', 1.5, 0.5, 2.1, 1),

('DCCuat', 1.0, 1.1, 1.3, 5),

('DCCinc', 1.4, 2.2, 2.4, 6),

('DCSeis', 2.2, 2.3, 1.3, 2),

('DCSiet', 0.7, 3.4, 0.3, 6),

('DCOcho', 2.4, 3.2, 1.9, 7);

GO

### -Inserción en tabla Aeropuerto

INSERT INTO Aeropuerto (codIATA, aeroNombre, aeroPais) VALUES

('MTV', 'Aeropuerto de MVD', 'Uruguay'),

('JFK', 'Aeropuerto de USA', 'United States'),

('EZE', 'Buenos Aires Ezeiza', 'Argentina'),

('SFO', 'Aeropuerto de SF', 'United States'),

('PEK', 'Aeropuerto Beijing', 'China'),

('SYD', 'Aeropuerto Sydney', 'Australia'),

('JNB', 'Aeropuerto Tambo', 'South Africa'),

('LHR', 'Aeropuerto Londres', 'United Kingdom');

GO

### -Inserción en tabla Carga

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-08-15', 500, 1, 'MTV', 'JFK', 'R'),

('P008', 'DCDos', '2023-08-15', 800, 2, 'JFK', 'EZE', 'T'),

('P009', 'DCTres', '2023-08-15', 200, 3, 'EZE', 'MTV', 'C'),

('P010', 'DCCuat', '2023-06-10', 500, 4, 'SYD', 'SFO', 'R'),

('P011', 'DCCinc', '2022-01-10', 100, 5, 'SFO', 'LHR', 'C'),

('P012', 'DCSeis', '2023-03-22', 200, 6, 'LHR', 'JNB', 'T'),

('P013', 'DCSiet', '2022-02-13', 150, 7, 'JNB', 'PEK', 'C'),

('P014', 'DCOcho', '2023-04-15', 300, 8, 'PEK', 'SYD', 'T'),

('P007', 'DCDos', '2023-07-12', 900, 1, 'MTV', 'EZE', 'C'),

('P008', 'DCTres', '2023-03-14', 100, 2, 'SFO', 'MTV', 'R'),

('P009', 'DCCuat', '2023-02-11', 700, 3, 'LHR', 'SFO', 'T'),

('P010', 'DCCinc', '2023-01-10', 550, 4, 'JNB', 'LHR', 'C'),

('P011', 'DCSeis', '2022-12-19', 50, 5, 'LHR', 'JNB', 'T'),

('P012', 'DCSiet', '2023-06-22', 1000, 6, 'JNB', 'JFK', 'C'),

('P013', 'DCOcho', '2022-09-12', 1500, 7, 'JFK', 'SYD', 'R'),

('P014', 'DCUno', '2023-05-25', 6000, 8, 'MTV', 'JFK', 'T'),

('P007', 'DCUno', '2023-04-01', 5550, 2, 'MTV', 'JFK', 'R'),

('P009', 'DCTres', '2023-01-30', 2120, 2, 'MTV', 'JFK', 'C'),

('P010', 'DCCuat', '2023-09-09', 5000, 2, 'SYD', 'PEK', 'R'),

('P011', 'DCCinc', '2022-01-07', 1001, 2, 'SYD', 'SFO', 'C'),

('P012', 'DCSeis', '2023-12-22', 800, 2, 'PEK', 'JNB', 'T'),

('P013', 'DCSiet', '2022-11-11', 760, 2, 'EZE', 'SYD', 'C'),

('P014', 'DCOcho', '2023-04-10', 630, 2, 'PEK', 'SYD', 'T'),

('P007', 'DCDos', '2023-07-01', 990, 2, 'MTV', 'EZE', 'C'),

('P009', 'DCCuat', '2023-02-19', 1100, 2, 'LHR', 'SFO', 'T'),

('P010', 'DCCinc', '2023-01-17', 5520, 2, 'SYD', 'LHR', 'C'),

('P011', 'DCSeis', '2022-12-29', 503, 2, 'LHR', 'JNB', 'T'),

('P012', 'DCSiet', '2023-06-30', 1040, 2, 'LHR', 'MTV', 'C'),

('P013', 'DCOcho', '2022-09-15', 1510, 2, 'JFK', 'SYD', 'R'),

('P014', 'DCUno', '2023-12-25', 6090, 2, 'MTV', 'JFK', 'T');

GO

# **2- Ejemplos Inválidos.**

**-Inserción en tabla Cliente**

**Cambio no válido:**

Agregamos un cuarto valor en la inserción, donde intentamos insertar un cliente con el mismo correo electrónico que ya existe, ejemplo: 'DiegoAlvezV@hotmail.com', esto viola la restricción que dice que el correo no puede repetirse.

INSERT INTO Cliente (cliNom, cliMail, cliCantCargas)

VALUES ('Diego Vaitcunas', 'DiegoAlvezV@hotmail.com', 2),

('Ignacio Pataro', 'IgnacioPataro2005@gmail.com', 0),

('Alberto Villar', 'AlbertoVillar1990@gmail.com', 5),

('Juan Perez', 'DiegoAlvezV@hotmail.com', 3);

### -Inserción en tabla avión

**Cambio no válido:**

Agregamos campos donde la capacidad del avión supera las cientocincuenta toneladas que se marca como límite en sus restricciones.

INSERT INTO Avion (avionID, avionMAT, avionMarca, avionModelo, avionCapacidad) VALUES

('P007', 'AB123', 'Boeing', '747', 180),

('P008', 'CD456', 'Pluna', 'A380', 220),

('P009', 'EF789', 'Emiratos Arabes', 'E190', 80);

### -Inserción en tabla Container

**Cambio no válido:**

En el último valor que insertamos, se intenta insertar un contenedor donde no cumple las reglas permitidas en la restricción que son 8 toneladas, porque la capacidad máxima no puede superar las 7 toneladas, los otros valores se mantienen dentro de los límites establecidos.

INSERT INTO DContainer (dContID, dContLargo, dContAncho, dContAlto, dContCapacidad) VALUES

('DCUno', 19, 9, 2.0, 900);

### 

### -Inserción en tabla Aeropuerto

**Cambio no válido:**

Intentamos insertar un código codAITA que tiene 4 caracteres y la restricción dice que tiene que tener exactamente 3 caracteres.

INSERT INTO Aeropuerto (codIATA, aeroNombre, aeroPais) VALUES

('MIAA', 'Miami International Airport', 'Estados Unidos');

### -Inserción en tabla Carga

**Cambio no válido:**

Insertamos un mismo valor en el campo dContId, esto no está permitido ya que tiene que ser único. Porque según la restricción este valor no puede repetirse para cada carga.

INSERT INTO Carga (idCarga, avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

(1, 'P007', 'DCUno', '2023-05-15', 500, 1, 'MTV', 'JFK', 'C'),

(2, 'P008', 'DCUno', '2023-05-17', 800, 2, 'JFK', 'EZE', 'R');

# **Consultas SQL**

### a. Mostrar los datos de los clientes que cargaron más kilos este año que el promedio total de kilos cargados por todos los clientes el año pasado:

SELECT c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

FROM Cliente c

JOIN Carga car ON car.cliID = c.cliID

WHERE YEAR(car.cargaFch) = YEAR(GETDATE())

GROUP BY c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

HAVING SUM(car.cargaKilos) > (

SELECT AVG(cargaKilos) FROM Carga

WHERE YEAR(cargaFch) = YEAR(GETDATE()) - 1);

### b. Del total de kilos cargados por cada avión, mostrar cuál fue el mayor valor, cuál fue el promedio y cuál fue el menor valor:

SELECT avionID, MAX(cargaKilos) AS MayorValor, AVG(cargaKilos) AS Promedio, MIN(cargaKilos) AS MenorValor

FROM Carga

GROUP BY avionID

order by MAX(cargaKilos);

### c. Para cada tipo de contenedor, mostrar sus datos, la cantidad de cargas en los que fue utilizado y el total de kilos cargados, si algún tipo de contenedor nunca fue utilizado, también deben mostrarse sus datos:

SELECT dc.dContID, dc.dContLargo, dc.dContAncho, dc.dContAlto, dc.dContCapacidad, COUNT(c.dContID) AS CargasUtilizadas, SUM(c.cargaKilos) AS TotalKilosCargados

FROM DContainer dc

LEFT JOIN Carga c ON dc.dContID = c.dContID

GROUP BY dc.dContID, dc.dContLargo, dc.dContAncho, dc.dContAlto, dc.dContCapacidad;

### d. Mostrar los datos de los clientes que utilizaron todos los aviones disponibles para sus cargas:

SELECT c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

FROM Cliente c

WHERE c.cliID IN (

SELECT cliID

FROM Carga

GROUP BY cliID

HAVING COUNT(DISTINCT avionID) = (SELECT COUNT(\*) FROM Avion)

);

### e. Mostrar el identificador de la carga, la fecha y los nombres de los aeropuertos de origen y destino para todas las cargas del año actual que utilizan aviones con una capacidad mayor a las 100 toneladas.

SELECT ca.idCarga, ca.cargaFch, aeroO.aeroNombre AS Origen, aeroD.aeroNombre AS Destino

FROM Carga ca

JOIN Aeropuerto aeroO ON ca.aeroOrigen = aeroO.codIATA

JOIN Aeropuerto aeroD ON ca.aeroDestino = aeroD.codIATA

WHERE YEAR(ca.cargaFch) = YEAR(GETDATE())

AND ca.avionID IN (

SELECT avionID

FROM Avion

WHERE avionCapacidad > 100

);

### f. Mostrar los datos del aeropuerto que recibió la mayor cantidad de kilos de los últimos 5 años.

SELECT TOP 1 ca.aeroDestino, SUM(ca.cargaKilos) AS TotalKilosRecibidos

FROM Carga ca

WHERE YEAR(ca.cargaFch) >= YEAR(GETDATE()) - 5

GROUP BY ca.aeroDestino

ORDER BY TotalKilosRecibidos DESC;

# **Utilizando T-SQL realizar los siguientes ejercicios:**

### a. Escribir un procedimiento almacenado que reciba como parámetros un rango de fecha y retorne también por parámetros el identificador de avión que cargó más kilos en dicho rango de fechas y el nombre del cliente que cargó más kilos en dicho rango (si hay más de uno, mostrar el primero).

CREATE PROCEDURE ProcedimientoA

@FechaInicio DATE,

@FechaFin DATE,

@AvionID CHAR(10) OUTPUT,

@ClienteNombre VARCHAR(50) OUTPUT

AS

BEGIN

SELECT TOP 1 @AvionID = c.avionID, @ClienteNombre = cl.cliNom

FROM Carga c

JOIN (

SELECT avionID, SUM(cargaKilos) AS TotalKilos

FROM Carga

WHERE cargaFch BETWEEN @FechaInicio AND @FechaFin

GROUP BY avionID

) AS CargasPorAvion ON c.avionID = CargasPorAvion.avionID

JOIN Cliente cl ON c.cliID = cl.cliID

WHERE c.cargaFch BETWEEN @FechaInicio AND @FechaFin

ORDER BY CargasPorAvion.TotalKilos DESC

END

/\*

DECLARE @AvionID CHAR(10), @ClienteNombre VARCHAR(50);

EXEC ProcedimientoA

@FechaInicio = '2020-01-01',

@FechaFin = '2024-01-01',

@AvionID = @AvionID OUTPUT,

@ClienteNombre = @ClienteNombre OUTPUT;

SELECT @AvionID AS AvionID, @ClienteNombre AS ClienteNombre;

\*/

### b. Realizar un procedimiento almacenado que, dadas las 3 medidas de un contenedor (largo x ancho x alto) retorne en una tabla los datos de los contenedores que coinciden con dichas medidas, de no existir ninguno se debe retornar un mensaje.

CREATE PROCEDURE ProcedimientoB

@largo Decimal,

@ancho Decimal,

@alto Decimal

AS

BEGIN

IF EXISTS (SELECT \* FROM DContainer WHERE dContLargo = @largo AND dContAncho = @ancho AND dContAlto = @alto)

BEGIN

SELECT dContID, dContLargo, dContAncho, dContAlto, dcontCapacidad

FROM DContainer

WHERE dContLargo = @largo AND dContAncho = @ancho AND dContAlto = @alto;

END

ELSE

BEGIN

SELECT 'No se encontraron contenedores con esas medidas' AS Mensaje;

END

END;

--Drop procedure ProcedimientoB

--EXEC ProcedimientoB @largo = 1.4, @ancho = 3.0, @alto = 2.0;

--EXEC ProcedimientoB @largo = 2.3, @ancho = 2.6, @alto = 2.3;

### c. Hacer una función que reciba un código de aeropuerto y retorne la cantidad de kilos recibidos de carga cuando ese aeropuerto fue destino.

CREATE FUNCTION FuncionC(@codigoAeropuerto VARCHAR(3))

RETURNS INT

AS

BEGIN

DECLARE @kilosRecibidos INT;

SELECT @kilosRecibidos = SUM(cargaKilos)

FROM Carga

WHERE aeroDestino = @codigoAeropuerto;

RETURN ISNULL(@kilosRecibidos, 0);

END;

--SELECT dbo.FuncionC('MTV') AS KilosRecibidos;

### d. Hacer una función que, para un cliente dado, retorne la cantidad total de kilos transportados por dicho cliente a aeropuertos de diferente país.

CREATE FUNCTION FuncionD( @clienteID INT)

RETURNS INT

AS

BEGIN

DECLARE @kilosTransportados INT;

SELECT @kilosTransportados = SUM(cargaKilos)

FROM Carga c

INNER JOIN Aeropuerto a ON c.aeroDestino = a.codIATA

WHERE c.cliID = @clienteID

AND a.aeroPais <> (SELECT aeroPais FROM Aeropuerto WHERE codIATA = c.aeroOrigen);

RETURN ISNULL(@kilosTransportados, 0);

END;

--SELECT dbo.FuncionD(1) AS KilosTransportados;

# **6. Escribir los siguientes disparadores :**

### a. Realizar un disparador que lleve un mantenimiento de la cantidad de cargas acumuladas de un cliente, este disparador debe controlar tanto los ingresos de cargas como el borrado de cargas.

CREATE TRIGGER TiggerA

ON Carga

AFTER INSERT, DELETE

AS

BEGIN

DECLARE @cliID INT;

IF EXISTS(SELECT \* FROM inserted)

BEGIN

SELECT @cliID = cliID FROM inserted;

END

ELSE IF EXISTS(SELECT \* FROM deleted)

BEGIN

SELECT @cliID = cliID FROM deleted;

END

UPDATE Cliente

SET cliCantCargas = (SELECT COUNT(\*) FROM Carga WHERE cliID = @cliID)

WHERE cliID = @cliID;

END;

/\*

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-08-12', 500, 1, 'MTV', 'JFK', 'R');

SELECT \* FROM Cliente WHERE cliID = 1;

DELETE FROM Carga WHERE cliID = 1;

SELECT \* FROM Cliente WHERE cliID = 1;\*/

### b. Hacer un disparador que, ante la modificación de cualquier medida de un contenedor, lleve un registro detallado en la tabla AuditContainer (ver estructura de la tabla en el anexo del presente obligatorio).

CREATE TRIGGER TiggerB

ON DContainer

AFTER UPDATE

AS

BEGIN

INSERT INTO AuditContainer (AuditFecha, AuditHost, LargoAnterior, AnchoAnterior, AltoAnterior, CapAnterior, LargoActual, AnchoActual, AltoActual, CapActual)

SELECT

GETDATE(),

HOST\_NAME(),

d.dContLargo,

d.dContAncho,

d.dcontAlto,

d.dcontCapacidad,

i.dContLargo,

i.dContAncho,

i.dcontAlto,

i.dcontCapacidad

FROM

inserted i

JOIN deleted d ON i.dContID = d.dContID

END;

UPDATE DContainer

SET dContLargo = 2.1, dContAncho = 3.4, dcontAlto = 1.5, dcontCapacidad = 2

WHERE dContID = 'DCOcho';

SELECT \*

FROM AuditContainer;

### c. Realizar un disparador que cuando se registra una nueva carga se valide que el avión tiene capacidad suficiente para almacenarla, esta verificación debe tener en cuenta todas las cargas que se están haciendo en ese avión en la misma fecha.

CREATE TRIGGER TriggerC

ON Carga

AFTER INSERT

AS

BEGIN

DECLARE @avionID int, @cargaFch datetime;

SELECT @avionID = avionID, @cargaFch = cargaFch

FROM inserted;

DECLARE @capacidadTotal decimal, @cargaTotal decimal;

SELECT @capacidadTotal = avionCapacidad, @cargaTotal = SUM(cargaKilos)

FROM Carga

JOIN Avion ON Carga.avionID = Avion.avionID

WHERE Carga.avionID = @avionID AND Carga.cargaFch = @cargaFch

GROUP BY Avion.avionCapacidad;

DECLARE @capacidadRestante decimal;

SET @capacidadRestante = @capacidadTotal - @cargaTotal;

IF @capacidadRestante < (SELECT cargaKilos FROM inserted)

BEGIN

RAISERROR ('Este avion no tiene capacidad suficiente para esa carga', 16, 1);

END;

END;

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-01-03', 700000, 1, 'MTV', 'JFK', 'R');

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-02-15', 70, 1, 'MTV', 'JFK', 'R');

SCRIPT Copy Paste

CREATE DATABASE DBCARGAS

GO

USE DBCARGAS

GO

-- Creacion de tablas SIN restricciones

CREATE TABLE Cliente(cliID int identity not null,

cliNom varchar(30) not null,

cliMail varchar(50),

cliCantCargas int)

GO

CREATE TABLE Avion(avionID char(10) not null,

avionMAT varchar(20) not null,

avionMarca varchar(30) not null,

avionModelo varchar(30) not null,

avionCapacidad decimal)

GO

CREATE TABLE Dcontainer(dContID char(6) not null,

dContLargo decimal,

dContAncho decimal,

dcontAlto decimal,

dcontCapacidad decimal)

GO

CREATE TABLE Aeropuerto(codIATA char(3) not null,

aeroNombre varchar(30) not null,

aeroPais varchar(30) not null)

GO

CREATE TABLE Carga(idCarga int identity not null,

avionID char(10) not null,

dContID char(3) not null,

cargaFch date,

cargaKilos decimal,

cliID int,

aeroOrigen char(3),

aeroDestino char(3),

cargaStatus char(1))

GO

CREATE TABLE AuditContainer(AuditID int identity not null,

AuditFecha datetime,

AuditHost varchar(30),

LargoAnterior decimal,

AnchoAnterior decimal,

AltoAnterior decimal,

CapAnterior decimal,

LargoActual decimal,

AnchoActual decimal,

AltoActual decimal,

CapActual decimal)

GO

-- Creación de todas las claves primarias.

ALTER TABLE Cliente

ADD CONSTRAINT PK\_Cliente PRIMARY KEY (cliID);

ALTER TABLE Avion

ADD CONSTRAINT PK\_Avion PRIMARY KEY (avionID);

ALTER TABLE DContainer

ADD CONSTRAINT PK\_DContainer PRIMARY KEY (dContID);

ALTER TABLE Aeropuerto

ADD CONSTRAINT PK\_Aeropuerto PRIMARY KEY (codIATA);

ALTER TABLE Carga

ADD CONSTRAINT PK\_Carga PRIMARY KEY (idCarga);

--Restricciones

--Email Unico

ALTER TABLE Cliente

ADD CONSTRAINT Unique\_Cliente\_Email UNIQUE (cliMail);

-- Otras restricciones relacionados a tabla Avion

-- Longitud maxima

ALTER TABLE Avion

ALTER COLUMN avionMAT varchar(20) not null;

-- Capacidad maxima avion

ALTER TABLE Avion

ADD CONSTRAINT CHK\_Avion\_Capacidad Check (avionCapacidad <= 150);

-- Capacidad maxima

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Largo CHECK (dContLargo <= 2.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Ancho CHECK (dContAncho <= 3.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Alto CHECK (dContAlto <= 2.5);

ALTER TABLE DContainer

ADD CONSTRAINT CHK\_DContainer\_Capacidad CHECK (dContCapacidad <= 7);

--Otras restricciones relacionadas a la tabla Carga.

ALTER TABLE Carga

ALTER COLUMN dContID char(6)

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAvion FOREIGN KEY (avionID) REFERENCES Avion(avionID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaDContainer FOREIGN KEY (dContID) REFERENCES DContainer(dContID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaCliente FOREIGN KEY (cliID) REFERENCES Cliente(cliID);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAeropuertoOrigen FOREIGN KEY (aeroOrigen) REFERENCES Aeropuerto(codIATA);

ALTER TABLE Carga

ADD CONSTRAINT FK\_CargaAeropuertoDestino FOREIGN KEY (aeroDestino) REFERENCES Aeropuerto(codIATA);

ALTER TABLE Carga

ADD CONSTRAINT UQ\_Carga\_avionID\_dContID\_cargaFch UNIQUE (avionID, dContID, cargaFch);

ALTER TABLE Carga

ADD CONSTRAINT CHK\_Carga\_Tipos CHECK (cargaStatus IN ('R', 'C', 'T', 'D', 'E'));

--Creación de índices que consideramos.

CREATE INDEX INDICE\_Cliente\_cliMail ON Cliente(cliMail);

CREATE INDEX INDICE\_Cliente\_cliCantCargas ON Cliente(cliCantCargas);

CREATE INDEX INDICE\_Avion\_avionID ON Avion(avionID);

CREATE INDEX INDICE\_DContainer\_dContID ON DContainer(dContID);

CREATE INDEX INDICE\_Aeropuerto\_codIATA ON Aeropuerto(codIATA);

CREATE INDEX INDICE\_Carga\_idCarga ON Carga(idCarga);

CREATE INDEX INDICE\_Carga\_avionID ON Carga(avionID);

CREATE INDEX INDICE\_Carga\_dContID ON Carga(dContID);

CREATE INDEX INDICE\_Carga\_cargaFch ON Carga(cargaFch);

CREATE INDEX INDICE\_Carga\_cliID ON Carga(cliID);

CREATE INDEX INDICE\_Carga\_aeroOrigen ON Carga(aeroOrigen);

CREATE INDEX INDICE\_Carga\_aeroDestino ON Carga(aeroDestino);

CREATE INDEX INDICE\_Carga\_cargaStatus ON Carga(cargaStatus);

--Ingreso completo de datos de prueba válidos.

INSERT INTO Avion (avionID, avionMAT, avionMarca, avionModelo, avionCapacidad) VALUES

('P007', 'AB123', 'Boeing', '747', 110),

('P008', 'CD456', 'Pluna', 'A380', 140),

('P009', 'EF789', 'Emiratos Arabes', 'E190', 80),

('P010', 'UV123', 'Airbus', 'A380', 140),

('P011', 'WX456', 'Boeing', '777', 150),

('P012', 'YZ789', 'Embraer', 'E190', 80),

('P013', 'AB234', 'Cessna', 'Citation CJ4', 6),

('P014', 'CD567', 'Bombardier', 'Global 6000', 10);

GO

INSERT INTO Cliente (cliNom, cliMail, cliCantCargas) VALUES

('Diego Vaitcunas', 'DiegoAlvezV@hotmail.com', 2),

('Ignacio Pataro', 'IgnacioPataro2005@gmail.com', 0),

('Alberto Villar', 'AlbertoVillar1990@gmail.com', 5),

('Sofía García', 'sofia.garcia@hotmail.com', 4),

('Gabriel Fernández', 'gabriel.fernandez@hotmail.com', 2),

('Valentina Ruiz', 'valentina.ruiz@gmail.com', 1),

('Lucas Morales', 'lucas.morales@hotmail.com', 3),

('Carolina Méndez', 'carolina.mendez@gmail.com', 0);

GO

INSERT INTO DContainer (dContID, dContLargo, dContAncho, dcontAlto, dcontCapacidad) VALUES

('DCUno', 2.3, 2.6, 2.3, 3),

('DCDos', 2.1, 3.4, 2.2, 4),

('DCTres', 1.5, 0.5, 2.1, 1),

('DCCuat', 1.0, 1.1, 1.3, 5),

('DCCinc', 1.4, 2.2, 2.4, 6),

('DCSeis', 2.2, 2.3, 1.3, 2),

('DCSiet', 0.7, 3.4, 0.3, 6),

('DCOcho', 2.4, 3.2, 1.9, 7);

GO

INSERT INTO Aeropuerto (codIATA, aeroNombre, aeroPais) VALUES

('MTV', 'Aeropuerto de MVD', 'Uruguay'),

('JFK', 'Aeropuerto de USA', 'United States'),

('EZE', 'Buenos Aires Ezeiza', 'Argentina'),

('SFO', 'Aeropuerto de SF', 'United States'),

('PEK', 'Aeropuerto Beijing', 'China'),

('SYD', 'Aeropuerto Sydney', 'Australia'),

('JNB', 'Aeropuerto Tambo', 'South Africa'),

('LHR', 'Aeropuerto Londres', 'United Kingdom');

GO

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-08-15', 500, 1, 'MTV', 'JFK', 'R'),

('P008', 'DCDos', '2023-08-15', 800, 2, 'JFK', 'EZE', 'T'),

('P009', 'DCTres', '2023-08-15', 200, 3, 'EZE', 'MTV', 'C'),

('P010', 'DCCuat', '2023-06-10', 500, 4, 'SYD', 'SFO', 'R'),

('P011', 'DCCinc', '2022-01-10', 100, 5, 'SFO', 'LHR', 'C'),

('P012', 'DCSeis', '2023-03-22', 200, 6, 'LHR', 'JNB', 'T'),

('P013', 'DCSiet', '2022-02-13', 150, 7, 'JNB', 'PEK', 'C'),

('P014', 'DCOcho', '2023-04-15', 300, 8, 'PEK', 'SYD', 'T'),

('P007', 'DCDos', '2023-07-12', 900, 1, 'MTV', 'EZE', 'C'),

('P008', 'DCTres', '2023-03-14', 100, 2, 'SFO', 'MTV', 'R'),

('P009', 'DCCuat', '2023-02-11', 700, 3, 'LHR', 'SFO', 'T'),

('P010', 'DCCinc', '2023-01-10', 550, 4, 'JNB', 'LHR', 'C'),

('P011', 'DCSeis', '2022-12-19', 50, 5, 'LHR', 'JNB', 'T'),

('P012', 'DCSiet', '2023-06-22', 1000, 6, 'JNB', 'JFK', 'C'),

('P013', 'DCOcho', '2022-09-12', 1500, 7, 'JFK', 'SYD', 'R'),

('P014', 'DCUno', '2023-05-25', 6000, 8, 'MTV', 'JFK', 'T'),

('P007', 'DCUno', '2023-04-01', 5550, 2, 'MTV', 'JFK', 'R'),

('P009', 'DCTres', '2023-01-30', 2120, 2, 'MTV', 'JFK', 'C'),

('P010', 'DCCuat', '2023-09-09', 5000, 2, 'SYD', 'PEK', 'R'),

('P011', 'DCCinc', '2022-01-07', 1001, 2, 'SYD', 'SFO', 'C'),

('P012', 'DCSeis', '2023-12-22', 800, 2, 'PEK', 'JNB', 'T'),

('P013', 'DCSiet', '2022-11-11', 760, 2, 'EZE', 'SYD', 'C'),

('P014', 'DCOcho', '2023-04-10', 630, 2, 'PEK', 'SYD', 'T'),

('P007', 'DCDos', '2023-07-01', 990, 2, 'MTV', 'EZE', 'C'),

('P009', 'DCCuat', '2023-02-19', 1100, 2, 'LHR', 'SFO', 'T'),

('P010', 'DCCinc', '2023-01-17', 5520, 2, 'SYD', 'LHR', 'C'),

('P011', 'DCSeis', '2022-12-29', 503, 2, 'LHR', 'JNB', 'T'),

('P012', 'DCSiet', '2023-06-30', 1040, 2, 'LHR', 'MTV', 'C'),

('P013', 'DCOcho', '2022-09-15', 1510, 2, 'JFK', 'SYD', 'R'),

('P014', 'DCUno', '2023-12-25', 6090, 2, 'MTV', 'JFK', 'T');

GO

/\*

-- Ejemplos Inválidos.

--Inserción en tabla Cliente

--INSERT INTO Cliente (cliNom, cliMail, cliCantCargas)

--VALUES ('Diego Vaitcunas', 'DiegoAlvezV@hotmail.com', 2),

--('Ignacio Pataro', 'IgnacioPataro2005@gmail.com', 0),

--('Alberto Villar', 'AlbertoVillar1990@gmail.com', 5),

--('Juan Perez', 'DiegoAlvezV@hotmail.com', 3);

--Inserción en tabla avión

--INSERT INTO Avion (avionID, avionMAT, avionMarca, avionModelo, avionCapacidad) VALUES

-- ('P007', 'AB123', 'Boeing', '747', 180),

-- ('P008', 'CD456', 'Pluna', 'A380', 220),

-- ('P009', 'EF789', 'Emiratos Arabes', 'E190', 80);

--Inserción en tabla Container

--INSERT INTO DContainer (dContID, dContLargo, dContAncho, dContAlto, dContCapacidad) VALUES

-- ('DCUno', 19, 9, 2.0, 900);

--Inserción en tabla Aeropuerto

--INSERT INTO Aeropuerto (codIATA, aeroNombre, aeroPais) VALUES

-- ('MIAA', 'Miami International Airport', 'Estados Unidos');

--Inserción en tabla Carga

--INSERT INTO Carga (idCarga, avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, --aeroDestino, cargaStatus) VALUES

-- (1, 'P007', 'DCUno', '2023-05-15', 500, 1, 'MTV', 'JFK', 'C'),

-- (2, 'P008', 'DCUno', '2023-05-17', 800, 2, 'JFK', 'EZE', 'R');

\*/

--Consultas SQL

-- a. Mostrar los datos de los clientes que cargaron más kilos este año que el promedio total de kilos cargados por todos los clientes el año pasado:

SELECT c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

FROM Cliente c

JOIN Carga car ON car.cliID = c.cliID

WHERE YEAR(car.cargaFch) = YEAR(GETDATE())

GROUP BY c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

HAVING SUM(car.cargaKilos) > (

SELECT AVG(cargaKilos) FROM Carga

WHERE YEAR(cargaFch) = YEAR(GETDATE()) - 1);

-- b. Del total de kilos cargados por cada avión, mostrar cuál fue el mayor valor, cuál fue el promedio y cuál fue el menor valor:

SELECT avionID, MAX(cargaKilos) AS MayorValor, AVG(cargaKilos) AS Promedio, MIN(cargaKilos) AS MenorValor

FROM Carga

GROUP BY avionID

order by MAX(cargaKilos);

-- c. Para cada tipo de contenedor, mostrar sus datos, la cantidad de cargas en los que fue utilizado y el total de kilos cargados, si algún tipo de contenedor

-- nunca fue utilizado, también deben mostrarse sus datos:

SELECT dc.dContID, dc.dContLargo, dc.dContAncho, dc.dContAlto, dc.dContCapacidad, COUNT(c.dContID) AS CargasUtilizadas, SUM(c.cargaKilos) AS TotalKilosCargados

FROM DContainer dc

JOIN Carga c ON dc.dContID = c.dContID

GROUP BY dc.dContID, dc.dContLargo, dc.dContAncho, dc.dContAlto, dc.dContCapacidad;

-- d. Mostrar los datos de los clientes que utilizaron todos los aviones disponibles para sus cargas:

SELECT c.cliID, c.cliNom, c.cliMail, c.cliCantCargas

FROM Cliente c

WHERE c.cliID IN (

SELECT cliID

FROM Carga

GROUP BY cliID

HAVING COUNT(DISTINCT avionID) = (SELECT COUNT(\*) FROM Avion)

);

-- e. Mostrar el identificador de la carga, la fecha y los nombres de los aeropuertos de origen y destino para todas las cargas del año actual

-- que utilizan aviones con una capacidad mayor a las 100 toneladas.

SELECT ca.idCarga, ca.cargaFch, aeroO.aeroNombre AS Origen, aeroD.aeroNombre AS Destino

FROM Carga ca

JOIN Aeropuerto aeroO ON ca.aeroOrigen = aeroO.codIATA

JOIN Aeropuerto aeroD ON ca.aeroDestino = aeroD.codIATA

WHERE YEAR(ca.cargaFch) = YEAR(GETDATE())

AND ca.avionID IN (

SELECT avionID

FROM Avion

WHERE avionCapacidad > 100

);

-- f. Mostrar los datos del que recibió la mayor cantidad de kilos de los últimos 5 años.

SELECT TOP 1 ca.aeroDestino, SUM(ca.cargaKilos) AS TotalKilosRecibidos

FROM Carga ca

WHERE YEAR(ca.cargaFch) >= YEAR(GETDATE()) - 5

GROUP BY ca.aeroDestino

ORDER BY TotalKilosRecibidos DESC;

--Utilizando T-SQL realizar los siguientes ejercicios:

--a. Escribir un procedimiento almacenado que reciba como parámetros un rango de fecha y retorne también por parámetros el identificador de avión que cargó más kilos en dicho

-- rango de fechas y el nombre del cliente que cargó más kilos en dicho rango (si hay más de uno, mostrar el primero).

--FUNCIONA--

CREATE PROCEDURE ProcedimientoA

@FechaInicio DATE,

@FechaFin DATE,

@AvionID CHAR(10) OUTPUT,

@ClienteNombre VARCHAR(50) OUTPUT

AS

BEGIN

SELECT TOP 1 @AvionID = c.avionID, @ClienteNombre = cl.cliNom

FROM Carga c

JOIN (

SELECT avionID, SUM(cargaKilos) AS TotalKilos

FROM Carga

WHERE cargaFch BETWEEN @FechaInicio AND @FechaFin

GROUP BY avionID

) AS CargasPorAvion ON c.avionID = CargasPorAvion.avionID

JOIN Cliente cl ON c.cliID = cl.cliID

WHERE c.cargaFch BETWEEN @FechaInicio AND @FechaFin

ORDER BY CargasPorAvion.TotalKilos DESC

END

/\*

DECLARE @AvionID CHAR(10), @ClienteNombre VARCHAR(50);

EXEC ProcedimientoA

@FechaInicio = '2020-01-01',

@FechaFin = '2024-01-01',

@AvionID = @AvionID OUTPUT,

@ClienteNombre = @ClienteNombre OUTPUT;

SELECT @AvionID AS AvionID, @ClienteNombre AS ClienteNombre;

\*/

--b. Realizar un procedimiento almacenado que, dadas las 3 medidas de un contenedor (largo x ancho x alto)

-- retorne en una tabla los datos de los contenedores que coinciden con dichas medidas, de no existir ninguno se debe retornar un mensaje.

--FUNCIONA--

CREATE PROCEDURE ProcedimientoB

@largo Decimal,

@ancho Decimal,

@alto Decimal

AS

BEGIN

IF EXISTS (SELECT \* FROM DContainer WHERE dContLargo = @largo AND dContAncho = @ancho AND dContAlto = @alto)

BEGIN

SELECT dContID, dContLargo, dContAncho, dContAlto, dcontCapacidad

FROM DContainer

WHERE dContLargo = @largo AND dContAncho = @ancho AND dContAlto = @alto;

END

ELSE

BEGIN

SELECT 'No se encontraron contenedores con esas medidas' AS Mensaje;

END

END;

--Drop procedure ProcedimientoB

--EXEC ProcedimientoB @largo = 1.4, @ancho = 3.0, @alto = 2.0;

--EXEC ProcedimientoB @largo = 2.3, @ancho = 2.6, @alto = 2.3;

--c. Hacer una función que reciba un código de aeropuerto y retorne la cantidad de kilos recibidos de carga cuando ese aeropuerto fue destino.

--FUNCIONA--

CREATE FUNCTION FuncionC(@codigoAeropuerto VARCHAR(3))

RETURNS INT

AS

BEGIN

DECLARE @kilosRecibidos INT;

SELECT @kilosRecibidos = SUM(cargaKilos)

FROM Carga

WHERE aeroDestino = @codigoAeropuerto;

RETURN ISNULL(@kilosRecibidos, 0);

END;

--SELECT dbo.FuncionC('MTV') AS KilosRecibidos;

--d. Hacer una función que, para un cliente dado, retorne la cantidad total de kilos transportados por dicho cliente a aeropuertos de diferente país.

--FUNCIONA--

CREATE FUNCTION FuncionD( @clienteID INT)

RETURNS INT

AS

BEGIN

DECLARE @kilosTransportados INT;

SELECT @kilosTransportados = SUM(cargaKilos)

FROM Carga c

INNER JOIN Aeropuerto a ON c.aeroDestino = a.codIATA

WHERE c.cliID = @clienteID

AND a.aeroPais <> (SELECT aeroPais FROM Aeropuerto WHERE codIATA = c.aeroOrigen);

RETURN ISNULL(@kilosTransportados, 0);

END;

--SELECT dbo.FuncionD(1) AS KilosTransportados;

--6. Escribir los siguientes disparadores :

--a. Realizar un disparador que lleve un mantenimiento de la cantidad de cargas acumuladas de un cliente,

-- este disparador debe controlar tanto los ingresos de cargas como el borrado de cargas.

--FUNCIONA--

CREATE TRIGGER TiggerA

ON Carga

AFTER INSERT, DELETE

AS

BEGIN

DECLARE @cliID INT;

IF EXISTS(SELECT \* FROM inserted)

BEGIN

SELECT @cliID = cliID FROM inserted;

END

ELSE IF EXISTS(SELECT \* FROM deleted)

BEGIN

SELECT @cliID = cliID FROM deleted;

END

UPDATE Cliente

SET cliCantCargas = (SELECT COUNT(\*) FROM Carga WHERE cliID = @cliID)

WHERE cliID = @cliID;

END;

/\*

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-08-12', 500, 1, 'MTV', 'JFK', 'R');

SELECT \* FROM Cliente WHERE cliID = 1;

DELETE FROM Carga WHERE cliID = 1;

SELECT \* FROM Cliente WHERE cliID = 1;\*/

-- b. Hacer un disparador que, ante la modificación de cualquier medida de un contenedor, lleve un registro detallado en la tabla AuditContainer

-- (ver estructura de la tabla en el anexo del presente obligatorio).

--FUNCIONA--

CREATE TRIGGER TiggerB

ON DContainer

AFTER UPDATE

AS

BEGIN

INSERT INTO AuditContainer (AuditFecha, AuditHost, LargoAnterior, AnchoAnterior, AltoAnterior, CapAnterior, LargoActual, AnchoActual, AltoActual, CapActual)

SELECT

GETDATE(),

HOST\_NAME(),

d.dContLargo,

d.dContAncho,

d.dcontAlto,

d.dcontCapacidad,

i.dContLargo,

i.dContAncho,

i.dcontAlto,

i.dcontCapacidad

FROM

inserted i

JOIN deleted d ON i.dContID = d.dContID

END;

drop trigger TiggerB;

UPDATE DContainer

SET dContLargo = 2.1, dContAncho = 3.4, dcontAlto = 1.5, dcontCapacidad = 2

WHERE dContID = 'DCOcho';

SELECT \*

FROM AuditContainer;

--c. Realizar un disparador que cuando se registra una nueva carga se valide que el avión tiene capacidad suficiente para almacenarla,

-- esta verificación debe tener en cuenta todas las cargas que se están haciendo en ese avión en la misma fecha.

--FATA TESTEAR--

CREATE TRIGGER TriggerC

ON Carga

AFTER INSERT

AS

BEGIN

DECLARE @avionID int, @cargaFch datetime;

SELECT @avionID = avionID, @cargaFch = cargaFch

FROM inserted;

DECLARE @capacidadTotal decimal, @cargaTotal decimal;

SELECT @capacidadTotal = avionCapacidad, @cargaTotal = SUM(cargaKilos)

FROM Carga

JOIN Avion ON Carga.avionID = Avion.avionID

WHERE Carga.avionID = @avionID AND Carga.cargaFch = @cargaFch

GROUP BY Avion.avionCapacidad;

DECLARE @capacidadRestante decimal;

SET @capacidadRestante = @capacidadTotal - @cargaTotal;

IF @capacidadRestante < (SELECT cargaKilos FROM inserted)

BEGIN

RAISERROR ('Este avion no tiene capacidad suficiente para esa carga', 16, 1);

END;

END;

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-01-03', 700000, 1, 'MTV', 'JFK', 'R');

INSERT INTO Carga (avionID, dContID, cargaFch, cargaKilos, cliID, aeroOrigen, aeroDestino, cargaStatus) VALUES

('P007', 'DCUno', '2023-02-15', 70, 1, 'MTV', 'JFK', 'R');