

EVALUACIÓN	Obligatorio	GRUPOS	todos	FECHA	Marzo 2023
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	<p>- Entrega:</p> <p>- Puntos: <u>Máximo:</u> 40. <u>Mínimo:</u> 1.</p> <p>IMPORTANTE</p> <p>- Los grupos deben estar conformados por hasta un máximo de tres personas.</p> <p>- Inscribirse (sacar la "<u>boleto de entrega</u>").</p>				

La realidad de este obligatorio referencia a un sistema de gestión de cargas aéreas de una empresa de envíos internacionales.



Una versión simplificada de un esquema de bases de datos para este sistema podría ser el siguiente:

Cliente(cliID, cliNom, cliMail, cliCantCargas)

Son los clientes que realizan envíos con la empresa de cargas, están identificados con un código de cliente auto incremental, se conoce su nombre, su correo electrónico (que no puede repetirse entre clientes) y la cantidad acumulada de cargas que lleva, cada vez que se concreta una carga se incrementa esta cantidad acumulada.

Avion(avionID, avionMAT, avionMarca, avionModelo, avionCapacidad)

Son los aviones donde se llevan las cargas, su identificador consta de 10 caracteres, se conoce la matrícula que puede tener letras y números, la marca y el modelo y la capacidad en toneladas que puede cargar, esta capacidad no puede superar las 150 toneladas.

DContainer(dContID, dContLargo, dContAncho, dContAlto, dContCapacidad)

Son los diferentes tipos de contenedor que utilizan las aeronaves para llevar la carga, están identificados con 3 letras y 3 números, se conoce su descripción y medidas, se sabe que el largo máximo no puede superar los 2.5mt, el ancho máximo no puede superar los 3.5m y la altura máxima no puede superar los 2.5mt y la capacidad máxima no puede superar las 7 toneladas.

Aeropuerto(codIATA,aeroNombre,aeroPais)

Son los aeropuertos donde se realizan las cargas de las aeronaves, están identificados por un código internacional de 3 caracteres, se conoce su nombre y su país.

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

Cada carga que es recibida en la aeronave se codifica con un número auto incremental, se sabe a que avión se carga, que tipo de contenedor lleva, la fecha (no es necesario registrar la hora), los kilos cargados, el cliente, los aeropuertos origen y destino, y el estado de la carga, el estado puede ser Reservado, Cargado, Transito, Descargado, Entregado.

Es importante controlar que los campos avionID, dContID y cargaFch no se pueden repetir para cada carga.

Se Pide:

1. Crear las restricciones de integridad que surjan del análisis de la letra, sobre el script de creación de tablas proporcionado (mediante ALTER de las estructuras dadas).
2. Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso).
3. Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos más que la cantidad). El mismo debería incluir ejemplos que deban ser rechazados por no cumplir con las restricciones implementadas.
4. Utilizando SQL implementar las siguientes consultas:
 - 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
 - b. Del total de kilos cargados por cada avión, mostrar cual fue el mayor valor, cual fue el promedio y cuál fue el menor valor.
 - 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.
 - d. Mostrar los datos de los clientes que utilizaron todos los aviones disponibles para sus cargas.
 - 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.
 - f. Mostrar los datos del aeropuerto que recibió la mayor cantidad de kilos de los últimos 5 años.

5. 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).
- 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.
- c. Hacer una función que reciba un código de aeropuerto y retorne la cantidad de kilos recibidos de carga cuando ese aeropuerto fue destino.
- d. Hacer una función que, para un cliente dado, retorne la cantidad total de kilos transportados por dicho cliente a aeropuertos de diferente país.

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.
- 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).
- 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.

Se debe entregar:

- a. Script con las restricciones de integridad creadas sobre el script de creación de tablas, índices, y el ingreso de datos de prueba.
- b. Script con la resolución de las consultas.
- c. Script con la resolución de los procedimientos y funciones
- d. Script con los disparadores

Consideraciones generales:

1. Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.
2. La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. **Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.**
3. Durante la última semana los docentes **no** contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.

Anexo 1 - Script de creación de tablas

```
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(3) 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
```