



Base de Datos

Control #3 (18-10-2016)

ALUMN@:

-
1. Considere el siguiente esquema de una base de datos de una empresa de almacenaje, ésta posee diferentes almacenes donde guarda cajas con productos de diferente tipo :

Almacen(Codigo:int, Ubicación: varchar(30), Capacidad:int)

Cajas(Numero:int, Contenido:varchar(40), Valor:int, Perecible:booleano, Cod_Almacen:int), donde Cod_Almacen es FK de Almacen, y perecible = true indica que el contenido de la caja es perecible.

Despacho(Num_Despacho:int, Numero_Caja:int, Id_Destino:int, fecha_despacho:date), donde Numero_Caja es FK de Cajas e Id_Destino es FK de Destino

Destino(Id_Destino:int, Descripción:varchar(30))

- a) Exprese en SQL los scripts necesarios para definir la tabla Despacho.
- b) Exprese en SQL los scripts necesario para ingresar las siguientes tuplas en la tabla Destino

100	Santiago
200	Osorno
300	Puerto Montt, Los Muermos

- c) Exprese en SQL los scripts necesarios para eliminar las tuplas en la tabla Almacen que tengan capacidad menor o igual a 0.
- d) Escriba una consulta en SQL que liste todos los destinos (descripción) que tienen despachos para después del 1-11-2016
- e) Escriba en SQL una consulta que entregue el código y ubicación del Almacen



donde hayan cajas con contenido perecible (use subconsultas).

- f) Obtener todos los números de caja y valor de cada una, de aquellas que tienen despacho para “Osorno”. (Use Join en su solución)
- g) Indicar código y ubicación de los almacenes que no contienen cajas con alimentos perecibles. (usar operador EXCEPT)
- h) Por cada almacén de la empresa mostrar el valor total de las cajas, el valor promedio de las cajas, el valor mínimo y el valor máximo, de aquellas con contenido perecible.
- i) Obtener código y ubicación de los almacenes que tienen más de 20 cajas para despachar al destino código 200.
- j) Encontrar el nombre de los destinos para los cuales no hay cajas con contenido perecible. (usar el operador EXISTS)



RESPUESTAS:

- a) Create Table Despacho(Num_Despacho integer, Numero_Caja integer, Id_Destino integer, fecha_despacho date, PRIMARY KEY (Num_Despacho), Foreign key (numero_caja) references cajas(numero), Foreign key (id_destino) references destino(id_destino)) ;
- b) INSERT INTO Destino VALUES (100, 'Santiago'), (200, 'Osorno'), (300,'Puerto Montt, Los Muermos');
- c) DELETE FROM Almacen WHERE capacidad <= 0;
- d) SELECT descripción from destino where id_destino IN (Select id_destino from despacho where fecha_despacho > '11-01-2016');
- e) SELECT codigo, ubicacion from Almacen where codigo IN (Select cod_almacen from cajas where perecible = true);
- f) SELECT Numero, valor from (Cajas Join Despacho on Numero = Numero_Caja) Natural Join Destino where descripción = 'Osorno';
- g) SELECT Codigo, ubicacion FROM Almacen where Codigo IN ((SELECT Cod_Almacen from Cajas) EXCEPT (SELECT Cod_Almacen from Cajas WHERE perecible = true));
- h) SELECT Cod_Almacen, sum(valor),avg(valor),min(valor),max(valor) FROM Cajas WHERE perecible = true GROUP BY Cod_Almacen;
- i) SELECT Codigo, Ubicacion FROM Almacen WHERE Codigo IN (SELECT Cod_Almacen FROM (Cajas JOIN Despacho On Numero = Numero_Caja) Where Id_Destino = 200 GROUP BY Cod_Almacen HAVING Count (*) > 20);
- j) SELECT descripción from destino D where NOT EXISTS (SELECT * from (despacho join cajas on numero = numero_caja) as C where perecible = true and D.id_destino = C.id_destino);