Instituto Tecnológico de Culiacán

Carrera: Ingeniería en Sistemas Computacionales

Materia: Temas Selectos de Bases de Datos

Alumnos:

José Alfredo García Aguilar

Cesar Alfredo Astorga Ochoa

Trabajo: U2 T2 Agregar Cities, Regiones y Countries a la BD Northwind

Fecha: 22-Marzo-2022

Horario de clase: 05:00 – 06:00 pm

Profesor: Daniel Esparza Soto

```
use NorthwindBD
go

--Agregar Cities, Regiones y Countries a la base de datos Northwind.
--Se propone los siguientes pasos:

--1.- Actualizar el campo Region que son nulos en las 3 tablas Employees, Customers
y Suppliers, actualizarlo como:  region = 'Region de ' + country
go
update employees set region ='Region de '+ country where region is null
go
update customers set region ='Region de '+ country where region is null
go
update suppliers set region ='Region de '+ country where region is null
go

--2.- Crear la tabla Countries ( CountyID , CountryName ) y llenarla con los datos
del campo Country de las tablas Employees, Customers y Suppliers.
go
create table countries (
countryid int identity (1,1) not null,
countryname nvarchar(50) not null
)
go

--Countries: Llave primaria
go
alter table countries add constraint pk_countries primary key(countryid)
--procedimiento almacenado para llenar Countries
go
create proc sp_llenarCountries @table nvarchar(50), @v nvarchar(1) as
declare @texto nvarchar(1000)
select @texto='insert into countries(countryname) ' + char(13) + 'select distinct
'+@v+'.country ' + char(13) + 'from '+@table+' '+@v + char(13) + ' where not exists
(select 1 from countries co where('+@v+'.country = co.countryname));'
exec sys.[sp_executesql] @texto --permite la sustitución de parámetros y es más
seguro y versátil que el execute
go
exec sp_llenarCountries employees,e
exec sp_llenarCountries customers,c
exec sp_llenarCountries suppliers,s
go

--3.- Crear la tabla Regiones ( RegionID, RegionName, CountryID ) y llenarla con los
datos del campo Region de las tablas Employees, Customers y Suppliers.
go
create table regiones (
regionid int identity (1,1) not null,
regionname nvarchar(50) not null,
countryid int not null
)
go
--Regiones: llave primaria
alter table regiones add constraint pk_regiones primary key(regionid)
go
--procedimiento almacenado para llenar Regiones
go
create proc sp_llenarRegiones @table nvarchar(50), @v nvarchar(1) as
```

```
declare @texto nvarchar(1000)
select @texto='insert into regiones(regionname,countryid) ' + char(13) + 'select
distinct '+@v+'.region,co.countryid ' + char(13) + 'from '+@table+' '+@v + char(13)
+  'inner join countries co on ' + @v + '.country = co.countryname ' + char(13) + '
where not exists (select 1 from regiones re where('+@v+'.region = re.regionname));'
exec sys.[sp_executesql] @texto --permite la sustitución de parámetros y es más
seguro y versátil que el execute
go
exec sp_llenarRegiones employees,e
exec sp_llenarRegiones customers,c
exec sp_llenarRegiones suppliers,s
go

--4.- Crear la tabla Cities (CityID, CityName, RegionID ) y llenarla con los datos
del campo City de las tablas Employees, Customers y Suppliers.
go
create table cities (
cityid int identity (1,1) not null,
cityname nvarchar(30) not null,
regionid int not null
)
go
--Cities: llave primaria
alter table cities add constraint pk_cities primary key(cityid)
go

insert into cities(cityname,regionid)
select city,regionid from employees e
inner join regiones r on r.regionname = e.region
union
select city,regionid from suppliers s
inner join regiones r on r.regionname = s.region
union
select city,regionid from customers c
inner join regiones r on r.regionname = c.region
group by city,regionid
go

--5.0 Crear las llaves externas entre Cities, Regiones y Countries
alter table cities add constraint fk_cities_regiones foreign key(regionid)--llave
foranea cities
references regiones(regionid)
go
alter table regiones add constraint fk_regiones_countries foreign key(countryid)--
llave foranea regiones
references countries(countryid)
go

--5.- Agregar a la tabla Customers el campo CityID.
go
alter table dbo.customers add cityid int null;
go

--6.- Actualizar el campo Customers.CityID con la clave CityID de la tabla Cities.
update customers set cityid = ci.cityid
from customers c
inner join cities ci on c.city = ci.cityname
go
```

```
--7.- Eliminamos los campos City,Region,Country en la tabla Customers .
go
alter table customers drop column country
go
drop index region on customers --sin el drop index no podemos eliminar las columnas
region ni city
go
alter table customers drop column region
go
drop index city on customers
go
alter table customers drop column city

--8.- Crear una llave externa de Customers con Cities.
alter table customers add constraint fk_customers_cities foreign key(cityid)
references cities(cityid)
go

--9.- Agregar a la tabla Employees el campo CityID.
go
alter table dbo.employees add cityid int null;
go

--10.- Actualizar el campo Employees.CityID con la clave CityID de la tabla Cities.
go
update employees set cityid = ci.cityid
from employees e
inner join cities ci on e.city = ci.cityname
go

--11.- Eliminamos los campos City,Region,Country en la tabla Employees .
go
alter table employees drop column country
go
alter table employees drop column region
go
alter table employees drop column city

--12.- Crear una llave externa de Employees con Cities.
alter table employees add constraint fk_employees_cities foreign key(cityid)
references cities(cityid)
go

--13.- Agregar a la tabla Suppliers el campo CityID.
go
alter table dbo.suppliers add cityid int null;
go

--14.- Actualizar el campo Suppliers.CityID con la clave CityID de la tabla Cities.
go
update suppliers set cityid =ci.cityid
from suppliers s
inner join cities ci on s.city = ci.cityname
go

--15.- Eliminamos los campos City,Region,Country en la tabla Suppliers .  >
go
```

```
alter table suppliers drop column country
go
alter table suppliers drop column region
go
alter table suppliers drop column city

--16.- Crear una llave externa de Suppliers con Cities.
alter table suppliers add constraint fk_suppliers_cities foreign key(cityid)
references cities(cityid)
go

--Selects para hacer las comprobaciones
select * from cities
select * from regiones
select * from countries
```