



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Laboratorio de:

SIC744 – Administración de Base de Datos

Práctica No.:

2

Grupo No.:

3

Integrantes:

Muñoz Gorky
Yaselga Ernesto

Tema:

Diseño e implementación de una Base de Datos

Objetivos:

- Realizar el diseño conceptual de un escenario propuesto
- Realizar el diseño lógico del modelo, uso de una herramienta case
- Implementación del modelo en un DBMS
- Creación de una base de datos
- Comprender e implementar la desnormalización

Marco teórico:

1. Diseño conceptual

Es la primera etapa en el diseño de una Base de Datos.

2. Modelo Entidad-Relación

Ideado por Peter Chen en 1976, el cual se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos. El modelo Entidad-Relación cuenta con los siguientes elementos:

- **Entidades:** son objetos, reales o abstractos, distinguibles de otros objetos de los cuales se pueda almacenar información en la base de datos. Existen dos tipos de entidades:
 - **Regulares:** son entidades normales que tienen existencia por si mismas sin depender de otras.
 - **Débiles:** son entidades que dependen de otras entidades para existir.
- **Atributos:** son las características de las entidades, es decir, las propiedades asociadas a un conjunto de entidades. Para cada atributo, existe un conjunto de valores permitidos, estos son llamados dominios.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

- **Relaciones:** son las conexiones o asociaciones entre diferentes entidades.
- **Cardinalidad:** indica el número de relaciones en las que una entidad puede aparecer.

3. Herramienta case

CASE son las siglas de *Computer-Aided Software Engineering*, lo que significa que es una herramienta de ayuda para un ingeniero de software para desarrollar y dar mantenimiento al software.

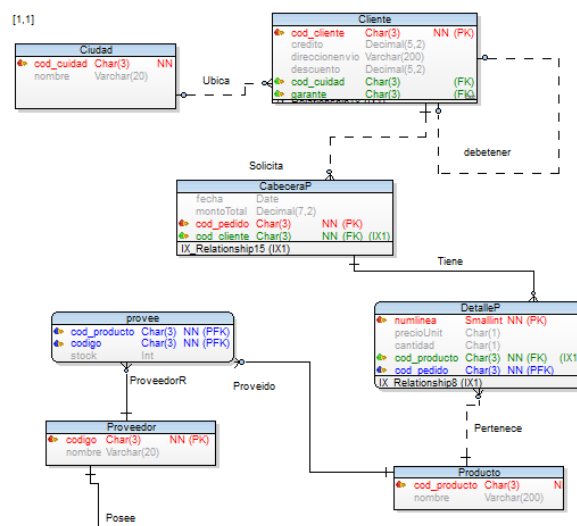
4. DBMS

DBMS son las siglas de *Dababase Magement System*, lo que se traduce a un sistema gestor de base de datos. Este es el software que brinda a los usuarios la capacidad de procesar, administrar y recuperar los datos que están almacenados en una base de datos. El DBMS está encargado de mantener la seguridad e integridad de los datos de la base de datos. Un DBMS cuenta con las siguientes herramientas:

- Herramienta para creación y especificación de la base de datos.
- Herramientas para la manipulación de los datos en la base de datos. Con esto permite realizar las operaciones como añadir, modificar, eliminar o consultar datos.
- Herramientas de recuperación de datos en caso de que existe algún inconveniente.
- Herramientas para la creación de copias de seguridad de la base de datos.

Desarrollo de la práctica:

1. Crear el modelo conceptual
2. Crear el modelo lógico con las restricciones de FK utilizando Toad Data Modeler, obtener el script de creación de tablas



3. Investigar cómo crear una Base de datos utilizando instrucciones de SQL, explique cada uno de los parámetros de la instrucción de creación de BD?



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

4. Explicar la utilidad de cada archivo que se crea en disco cuando se crea un Base de datos en SQL Server.

Se crean 3 tipos de archivos:

- **Primario:** se crea con la base de datos y almacena datos y objetos de usuario.
- **Secundario:** es un archivo opcional el cual se usa para distribuir los datos en diferentes discos, también sirve para aumentar la capacidad de la BD.
- **Log:** Guarda registros que son útiles para recuperar la base de datos, toda las de datos debe tener almenos un archivo de log.

5. Crear la base de datos pedidos con las siguientes condiciones:

- Dividir la BD en dos archivos (uno corresponde al filegroup primary y el otro a un filegroup secundario): en el secundario se colocarán las tablas que están sujetas a pocas o ninguna modificación, como datos de los Clientes, Fábricas y Productos; en el primario las demás tablas.
- Los archivos de datos y los archivos log deben estar en diferentes particiones (Utilizar las carpetas creadas durante la instalación del DBMS).
- El tamaño inicial de cada archivo de datos es de 5 MB, con un incremento automático del 10% y un límite de 10MB.
- El tamaño inicial de los archivos log es de 2MB, con un incremento automático de 1 MB y un límite de crecimiento a 8 MB.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

SQLQuery1.sql - Wl...S6.master (sa (52))

```
use master;
go
create database pedidos
on primary
( name = pedido1,
  filename = 'D:\Data\pedido1.mdf',
  size = 5,
  maxsize = 10,
  filegrowth = 10%),
FILEGROUP Secundario
( name = pedido2,
  filename = 'D:\Data\pedido2.mdf',
  size = 5,
  maxsize = 10,
  filegrowth = 10%),
( name = pedido3,
  filename = 'D:\Data\pedido3.mdf',
  size = 5,
  maxsize = 10,
  filegrowth = 10%)
log on
(name = pedido1_log,
 filename = 'D:\Log\pedidolog.idf',
 size = 2MB,
 maxsize = 10MB,
 filegrowth = 10%)
```

91 %

Messages

Commands completed successfully.

Database Properties - pedidos

Select a page: General, Files, Filegroups, Options, Change Tracking, Permissions, Extended Properties, Mirroring, Transaction Log Shipping, Query Store

Connection

Server: WIN-86795B91536
Connection: sa
[View connection properties](#)

Progress

Ready

Database name: pedidos
Owner: sa
☒ Use full-text indexing

Database files:

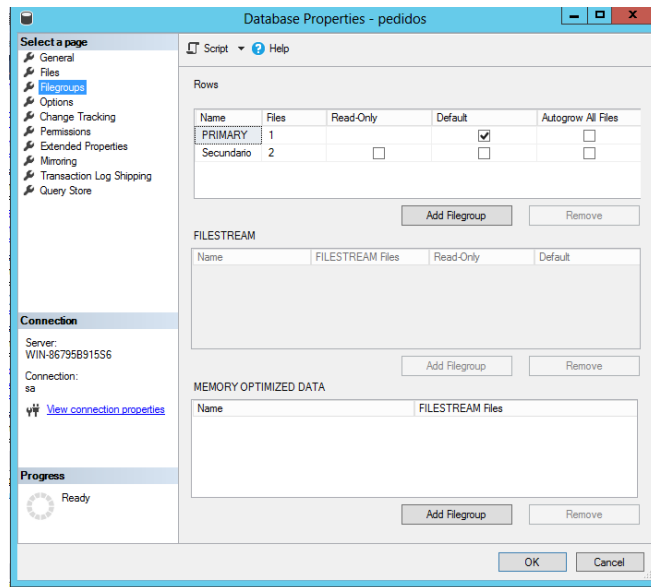
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
pedido1	ROWS...	PRIMARY	8	By 10 percent, Limited to 10 ...
pedido2	ROWS...	Secundario	5	By 10 percent, Limited to 10 ...
pedido3	ROWS...	Secundario	5	By 10 percent, Limited to 10 ...
pedido1_log	LOG	Not Applicable	2	By 10 percent, Limited to 10 ...

Add Remove

OK Cancel



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



6. Modificar el script generado en el punto 2 de la siguiente manera:
- Comentar la generación de Integridad referencial (FK)
 - Crear las tablas en los filegroups definidos en el punto 5
 - Crear dos esquemas: movimientos y catálogo; en catálogo se colocarán las tablas de clientes, fábricas y productos, las restantes se colocarán en movimientos.

```
File Edit Selection Find View Goto Tools Project Preferences Help File Edit Selection Find View Goto Tools Project Preferences Help
labo2SQL untitled untitled labo2SQL untitled untitled Scrip-la

145 -- Table Telefono
146
147 CREATE TABLE [Telefono]
148 (
149     [numero] Char(10) NOT NULL,
150     [codigo] Char(3) NOT NULL
151 )
152 go
153
154 -- Add keys for table Telefono
155
156 --ALTER TABLE [Telefono] ADD CONSTRAINT [PK_Tele
157 --go
158
159 -- Table provee
160
161 CREATE TABLE [provee]
162 (
163     [cod_producto] Char(3) NOT NULL,
164     [codigo] Char(3) NOT NULL,
165     [stock] Int NULL
166 )
167 go
168 /*
169 -- Add keys for table provee
170
171 ALTER TABLE [provee] ADD CONSTRAINT [PK_provee]
172 go
173
174 -- Create foreign keys (relationships) section -
175
176
177 ALTER TABLE [Telefono] ADD CONSTRAINT [Posee] FO
178 go
179
180 ALTER TABLE [DetalleP] ADD CONSTRAINT [Pertenece
181 go
182

209 use pedidos
210 go
211 create schema catalogo
212 go
213 create schema movimiento
214 go
215 CREATE TABLE [catalogo].[Ciudad]
216 (
217     [cod_ciudad] Char(3) NOT NULL,
218     [nombre] Varchar(20) NULL
219 )
220 ON [Secundario]
221 go
222
223 CREATE TABLE [catalogo].[Cliente]
224 (
225     [cod_cliente] Char(3) NOT NULL,
226     [credito] Decimal(5,2) NULL,
227     --CONSTRAINT [chk-credito] CHECK (credito<2000),
228     [direccionenvio] Varchar(200) NULL,
229     [descuento] Decimal(5,2) NULL,
230     --CONSTRAINT [chk-descuento] CHECK (0 < descuento < 100),
231     [cod_ciudad] Char(3) NULL,
232     [garante] Char(3) NULL
233 )
234 ON [Secundario]
235 go
236
237 CREATE TABLE [catalogo].[Producto]
238 (
239     [cod_producto] Char(3) NOT NULL,
240     [nombre] Varchar(200) NULL
241 )
242 ON [Secundario]
243 go
244
245 CREATE TABLE [movimiento].[CabeceraP]
246
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

7. Desnormalización: Crear una tabla cabezacuerpoP que unifique en una sola tabla los datos de las tablas CabezaP y CuerpoP. Las tablas originales deben estar sincronizadas con cabezacuerpoP, de tal forma que si se modifican los datos de las originales, se cambien los datos de la tabla cabezacuerpoP o viceversa, ¿cómo hacerlo?

```
CREATE TABLE [movimiento].[CuerpoCabeceraP]
(
    [cod_producto] Char(3) NOT NULL,
    [cod_pedido] Char(3) NOT NULL,
    [cod_cliente] Char(3) NOT NULL,
    [fecha] Date NULL,
    [montoTotal] Money NULL,
    [numlinea] Smallint NOT NULL,
    [precioUnit] Money NULL,
    [cantidad] Int NULL,
)
ON [Primary]
go
```

```

)
ON [Primary]
go

create procedure sp_insertar_cuerpocabecera
@AC_cod_producto Char(3),
@AC_cod_pedido Char(3),
@AC_cod_cliente Char(3),
@AD_fecha Date,
@AM_montoTotal Money,
@AI_numlinea Smallint,
@AM_precioUnit Money,
@AI_cantidad Int
AS
BEGIN
    IF EXISTS (SELECT TOP 1 1 FROM [movimiento].[CuerpoCabeceraP] WHERE cod_pedido = @AC_cod_pedido and numlinea = @AI_numlinea)
    BEGIN
        UPDATE [movimiento].[CuerpoCabeceraP]
        SET
            cod_producto = @AC_cod_producto,
            cod_pedido = @AC_cod_pedido,
            cod_cliente = @AC_cod_cliente,
            fecha = @AD_fecha,
            montoTotal = @AM_montoTotal,
            numlinea = @AI_numlinea,
            precioUnit = @AM_precioUnit,
            cantidad = @AI_cantidad
        WHERE cod_pedido = @AC_cod_pedido and numlinea = @AI_numlinea
    END
    ELSE
    BEGIN
        insert into [movimiento].[CuerpoCabeceraP] values (@AC_cod_producto, @AC_cod_pedido, @AC_cod_cliente, @AD_fecha, @AM_montoTotal, @AI_numlinea, @AM_precioUnit, @AI_cantidad)
    END
END
```

```
CREATE TRIGGER [movimiento].[tg_insertar_DETALLEP]
on [movimiento].[DETALLEP]
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @LV_codPed char(3), @LD_fecha DATE, @LM_montoTotal MONEY, @LV_codCli CHAR(3), @LI_numlinea INT, @LM_precioU MONEY, @LI_cantidad INT, @LV_codPro CHAR(3)
    DECLARE Cursor1 CURSOR FOR SELECT inserted.cod_pedido, fecha, montoTotal, cod_cliente, numLinea, precioUnit, cantidad, cod_producto FROM inserted, [movimiento].[CABECERA]
    OPEN Cursor1
    FETCH NEXT FROM Cursor1 INTO @LV_codPed, @LD_fecha, @LM_montoTotal, @LV_codCli, @LI_numlinea, @LM_precioU, @LI_cantidad, @LV_codPro
    WHILE @@fetch_status = 0
    BEGIN
        exec [dbo].[sp_insertar_cuerpocabecera] @AC_cod_pedido = @LV_codPed, @AD_fecha = @LD_fecha, @AM_montoTotal = @LM_montoTotal, @AC_cod_cliente = @LV_codCli, @AI_numlinea = @LI_numlinea, @AM_precioUnit = @LM_precioU, @AI_cantidad = @LI_cantidad, @LV_codPro = @LV_codPro
    END
    FETCH NEXT FROM Cursor1 INTO @LV_codPed, @LD_fecha, @LM_montoTotal, @LV_codCli, @LI_numlinea, @LM_precioU, @LI_cantidad, @LV_codPro
    CLOSE Cursor1
    DEALLOCATE Cursor1
END
go
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

```
INSERT INTO [movimiento].[DETALLEP]  
values(3,10.5,5,'P02','PD4')  
select * from [movimiento].[CuerpoCabeceraP]
```

9 %

Results Messages

	cod_producto	cod_pedido	cod_cliente	fecha	montoTotal	numlinea	precioUnit	cantidad
1	P05	PD1	C07	2018-10-01	500,00	3	10,50	5
2	P02	PD4	C03	2018-10-04	500,00	3	10,50	5

Los datos se ingresa solo en detalle y el trigger se encarga de empatar con la cabecera.

Conclusiones y recomendaciones:

- Se logró realizar el diseño conceptual de un escenario propuesto.
- Se realizó el diseño lógico del modelo en una herramienta case llamada "Toad Data Modeler"
- Se logró la creación de una base de datos en base a los requerimientos.
- Se logró crear FILEGROUPS en diferentes directorios dentro del computador.
- Se creó tablas en diferentes esquemas de la base.
- Se logró implementar un trigger para de esta forma aplicar la desnormalización.
- Se recomienda tener previamente creadas las carpetas de DATA, LOG y BACKUP
- Se recomienda no usar punteros ya que afectan a la eficiencia de las sentencias.

Bibliografía:

- [1] Universidad Politécnica de Valencia. "Introducción a Herramientas CASE y System Architect", [Online] disponible en: http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf
- [2] Sánchez J. "Diseño conceptual de base de datos" (2004). [Online] disponible en: <http://www.emtelco.com.co/sites/default/files/2017-03/disenobd.pdf>
- [3] Universidad de Granada. "Diseño conceptual", [Online] disponible en: <https://elvex.ugr.es/idbis/db/docs/design/3-conceptual.pdf>