

Fundamentos de Bases de Datos.

Práctica 5.

Profesor: M.I. Gerardo Avilés Rosas

gar@ciencias.unam.mx

Laboratorio: Luis Eduardo Castro Omaña

lalo_castro@ciencias.unam.mx

23 de septiembre de 2019

Se dan a conocer especificaciones de entrega para la practica 5.

1. Modelo Relacional

El modelo relacional se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos. Esto debido a su simplicidad, que facilita el trabajo del programador en comparación con otros modelos.

1.1. Estructura básica de una base relacional

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional.

Una tabla de n atributos debe ser un subconjunto de

$$D_1 \times D_2 \times \dots \times D_{n-1} \times D_n \quad (1)$$

Los matemáticos definen las relaciones como subconjuntos del producto cartesiano de la lista de dominios. Esta definición se corresponde de manera casi exacta con la definición de tabla dada anteriormente. La única diferencia es que aquí se han asignado nombres a los atributos, mientras que los matemáticos sólo utilizan «nombres» numéricos, utilizando el entero 1 para denotar el atributo cuyo dominio aparece en primer lugar en la lista de dominios, 2 para el atributo cuyo dominio aparece en segundo lugar, etcétera. Como las tablas son

esencialmente relaciones, se utilizarán los términos matemáticos relación y tupla en lugar de los términos tabla y fila. Una variable tupla es una variable que representa a una tupla; en otras palabras, una tupla que representa al conjunto de todas las tuplas.

El orden en que aparecen las tuplas es irrelevante, dado que una relación es un conjunto de tuplas. Así, si las tuplas de una relación se muestran ordenadas o desordenadas, no importa; las relaciones de estas figuras son las mismas, ya que ambas contienen el mismo conjunto de tuplas.

Para cada atributo, de una relación, hay un conjunto de valores permitidos, llamado dominio. Para todas las relaciones, el dominio de todos los atributos de r deben ser atómicos. Un dominio es atómico si los elementos del dominio se consideran unidades indivisibles. Es posible que varios atributos tengan el mismo dominio. Un valor de dominio que es miembro de todos los dominios posibles es el valor nulo, que indica que el valor es desconocido o no existe.

1.2. Conversión de un diagrama E/R a tablas

Un esquema de bases de datos, junto con las dependencias de clave primaria y externa, se puede mostrar gráficamente mediante diagramas de tablas. Cada relación aparece como un cuadro con los atributos listados dentro de él y el nombre de la relación sobre él. Si hay atributos clave primaria, una línea horizontal cruza el cuadro con los atributos clave primaria listados sobre ella. Las dependencias de clave externa aparecen como flechas desde los atributos clave externa de la relación referenciante a la clave primaria de la relación referenciada.

Para realizar la conversión de un diagrama E/R a un diagrama de tablas se debe seguir los siguiente:

- Una entidad fuerte se convierte en una tabla con los mismos atributos.
- Una entidad débil se convierte en tabla con los atributos que contenga la entidad, además de agregar los atributos marcados como llave de la entidad fuerte de la que depende la entidad.
- Una relación (en el modelo E/R) se convierte en tabla, donde sus atributos serán los atributos marcados como llaves primarias en las entidades que relaciona. aquí debemos considerar las siguiente posibilidades:
 - Relaciones 1:1 existen la siguiente posibilidades:
 1. Relación parcial: La relación se convierte en tabla en la cual están las llaves primarias de la entidades que relaciona.
 2. Relación total de un lado: Crear una tabla con todos los atributos de la entidad que tiene la relación total e incluir los atributos de la relación, así como las llaves primarias de la entidad con la relación parcial.

3. Relación total de ambos lados: Crear una tabla con todos los atributos de la relación y los atributos de las entidades que estén relacionadas.
- Relaciones 1:n. Se crea una tabla con todos los atributos de la entidad con cardinalidad n y se agregan los atributos de la relación y los atributos llave de la entidad con cardinalidad 1.
 - Relaciones n:n. Similar a los casos 1:1. La relación se convierte en tabla y se agregan los atributos llaves de las entidades que relacione.
- Los atributos multivaluados se convierten en tablas, los atributos de la tabla serán el atributo multivaluado y la llave de la entidad a la que corresponde le atributo multivaluado.
 - Especialización-generalización total, parcial, disyunción o traslape. Se crea una tabla para la super-entidad con todos sus atributos y se crea una tabla T_i para cada sub-entidad que contenga los atributos de esa sub-entidad más el atributo llave de la entidad superior.

2. Creación Base de Datos en SQL Server

En SQL Server existen diversas formas de crear una base de datos. Una opción para la creación es hacer uso de la herramienta SSMS, otra alternativa es realizar scripts con operaciones DDL y hacer uso de T-SQL para la creación de las bases de datos.

2.1. T-SQL

T-SQL es la versión de SQL Server para SQL (Structured Query Language) lenguaje estándar para realizar queries a las base de datos. SQL es utilizado por la mayoría de los RDBMS (Relational DataBase Management System).

Existen dos tipos de instrucciones en SQL. Las instrucciones DDL (Data Definition Language) encargadas de la creación, modificación o eliminación de objetos en el RDBMS como bases de datos, tablas y procedimientos almacenados. Por otro lado, las instrucciones DML (Data Manipulation Language) son las instrucciones que trabajan únicamente con datos (select, insert, update, delete).

2.2. Creación de Base de Datos con T-SQL

La instrucción DDL para la creación de una base de datos es:

```
CREATE DATABASE [NOMBRE_BD];
```

Es necesario que el nombre de la base sea único, de otra manera no será posible crear la base de datos.

La instrucción mostrada anteriormente crea una base de datos con los valores por default definidos en la base de datos Model. Para poder crear una base de datos mas personalizada es necesario agregar algunos parámetros de configuración en la instrucción como se muestra a continuación.

```
CREATE DATABASE [NOMBRE_BD]
ON PRIMARY
(
NAME = [NOMBRE_ARCHIVO_LOGICO],
FILENAME = '$PATH\[NOMBRE_ARCHIVO_LOGICO].mdf',
SIZE = [size [ KB | MB | GB | TB ]],
MAXSIZE = max_size [ KB | MB | GB | TB ] | UNLIMITED },
FILEGROWTH = TAMAÑO_INCREMENTO [ KB | MB | GB | TB | % ]
)
LOG ON
(
NAME = [NOMBRE_ARCHIVO_LOGICO]_Log,
FILENAME = '$PATH\[NOMBRE_ARCHIVO_LOGICO]_Log.ldf',
SIZE = [size [ KB | MB | GB | TB ]],
MAXSIZE = max_size [ KB | MB | GB | TB ] | UNLIMITED },
FILEGROWTH = TAMAÑO_INCREMENTO [ KB | MB | GB | TB | % ]
);
```

Se puede encontrar mas configuración en la documentación oficial de SQL Server.

2.3. Fundamentos de Data Base Scripting

El propósito de crear scripts para la creación de bases de datos es poder reutilizar los scripts para futura creación de bases. Esta reusabilidad permite, por ejemplo, crear bases de datos en múltiples ambientes.

Para asegurar la funcionalidad de nuestro scripts es necesario verificar algunos escenarios:

- Verificar si la base de datos existe.
- Si la base no existe, crearla.
- Si no existe, no realizar alguna operación.

Para verificar si la base de datos existe es necesario utilizar la tablas de **sys** de la base de datos Resources, la cual contiene la información del servidor y bases de datos,

```
USE Master;
IF NOT EXISTS (SELECT 1 FROM sys.databases WHERE [name] = '[NOMBRE_NUEVA_BD]')
```

```

BEGIN
CREATE DATABASE NOMBRE_NUEVA_BD
ON PRIMARY
(
NAME = [NOMBRE_ARCHIVO_LOGICO],
FILENAME = '$PATH\[NOMBRE_ARCHIVO_LOGICO].mdf',
SIZE = [size [ KB | MB | GB | TB ]],
MAXSIZE = max_size [ KB | MB | GB | TB ] | UNLIMITED },
FILEGROWTH = TAMAÑO_INCREMENTO [ KB | MB | GB | TB | % ]
)
LOG ON
(
NAME = [NOMBRE_ARCHIVO_LOGICO]_Log,
FILENAME = '$PATH\[NOMBRE_ARCHIVO_LOGICO]_Log.ldf',
SIZE = [size [ KB | MB | GB | TB ]],
MAXSIZE = max_size [ KB | MB | GB | TB ] | UNLIMITED },
FILEGROWTH = TAMAÑO_INCREMENTO [ KB | MB | GB | TB | % ]
);
END;
GO

```

2.4. Eliminar base de datos

Rollbacks scripts son necesarios para revertir los cambios realizados anteriormente. Son bastantes útiles para conservar el estado actual de la base de datos, lo que ayuda a mitigar errores introducidos por la creación de nuevos objetos.

El siguiente comando elimina una base de datos, verificando que la base de datos exista para evitar errores en la ejecución del script:

```

USE Master;
IF EXISTS (SELECT 1 FROM sys.databases WHERE [name] = '[NOMBRE_BD_ELIMINAR]')
BEGIN
DROP DATABASE NOMBRE_BD_ELIMINAR;
END;
GO

```

3. Actividad

3.1. Modelo Realcional

Deberán instalar Visual Paradigm en la edición Community disponible aquí: <https://www.visual-paradigm.com/download/community.jsp>

Deberán realizar la conversión del diagrama E-R que realizaron en la práctica pasada a un diagrama relacional utilizando Visual Paradigm.

Sus diagramas deben de tener las referencias entre las relaciones de manera correcta. Deben especificar el dominio de sus atributo y sus restricciones si es que existen.

Deben especificar las llaves foráneas, compuestas y primarias de cada relación.

3.2. Creación base de datos:

Se debe crear un script de T-SQL para la creación de una base de datos. La base de datos debe seguir la siguiente configuración:

- El nombre de la base de datos debe ser *FBD2020-1*- seguido de los últimos cuatro números de cuenta del alumno que envía la práctica. Ejemplo: *FBD2020-1-5669*.
- Un data file de 10MB de tamaño. El path debe ser el siguiente */fbd/fundamentos*
- Un log file de 2 MB de tamaño. El path debe ser el siguiente */fbd/fundamentos*
- El data file se debe auto incrementar un 50 % y puede ser tan grande como lo necesite.
- El data log de debe auto incrementar 2MB en tamaño pero no puede crecer mas de 100MB.
- Se debe verificar que no exista la base de datos. Si existe se debe eliminar la base existente y crearla nuevamente.

4. Entregables

Deberán agregar su archivo *.dia* de la practica pasada en la carpeta doc, así como el archivo *.vpp* que contiene el diagrama relacional. Recuerda que si están trabajando en equipo es necesario que cada vez que se reúnan para discutir el diseño de la base deberán generar una bitácora sobre lo discutido en esa sesión. Deberán agregar dicha bitácora en su reporte.

Deberán hacer una descripción de las relaciones que resulten de la conversión, explicando las referencias de sus llaves y por qué razón le asignaron el dominio a cada atributo de la relación.

El reporte debe contener la conversión del modelo E-R, las restricciones del modelo, la descripción de las relaciones y las bitácoras.

Por ultimo dentro de la carpeta sql/DDL se debe insertar el script para la creación de la base de datos.

La entrega deberá ser el día lunes 30 de Septiembre de 2019.