

Fundamentos de Bases de Datos.

Práctica 4.

Profesor: M.I. Gerardo Avilés Rosas
gar@ciencias.unam.mx
Laboratorio: Luis Eduardo Castro Omaña
lalo_castro@ciencias.unam.mx

9 de septiembre de 2019

Se dan a conocer especificaciones de entrega para la practica 4.

1. Modelado de datos

Aunque los conceptos básicos de E-R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser más adecuadamente expresados mediante ciertas extensiones del modelo E-R básico. A continuación se describen las características del modelo E-R extendido.

1.1. Modelo entidad-relación extendido

Los elementos del modelo E-R extendido son:

1.1.1. Especialización

Un conjunto de entidades puede incluir subgrupos de entidades que se diferencian de alguna forma de las otras entidades del conjunto. Por ejemplo, un subconjunto de entidades en un conjunto de entidades puede tener atributos que no son compartidos por todas las entidades del conjunto de entidades. El proceso de designación de subgrupos dentro de un conjunto de entidades se denomina especialización.

En términos de un diagrama E-R, la especialización se representa mediante un componente triangular etiquetado ES. La relación ES se puede llamar también relación superclase-subclase. Los conjuntos de entidades de nivel más alto y más bajo se representan como conjuntos de entidades regulares, es decir, como rectángulos que contienen el nombre del conjunto de entidades.

La especialización parte de un conjunto de entidades simple; enfatiza las diferencias entre las entidades dentro del conjunto mediante la creación de distintos conjuntos de entidades de nivel más bajo. Estos conjuntos de entidades de nivel más bajo pueden tener atributos, o pueden participar en relaciones que no se aplican a todas las entidades del conjunto de entidades de nivel más alto. Realmente, la razón de que el diseñador aplique la especialización es representar tales características diferentes.

1.1.2. Generalización

El refinamiento a partir de un conjunto de entidades inicial en sucesivos niveles de subgrupos de entidades representa un proceso de diseño descendente en el que las distinciones se hacen explícitas. El proceso de diseño puede ser también de una forma ascendente, en el que varios conjuntos de entidades se sintetizan en un conjunto de entidades de nivel más alto basado en características comunes.

La generalización, es una relación contenedora que existe entre el conjunto de entidades de nivel más alto y uno o más conjuntos de entidades de nivel más bajo. Los conjuntos de entidades de nivel más alto y nivel más bajo también se pueden llamar superclase y subclase, respectivamente. Para todos los propósitos prácticos, la generalización es una inversión simple de la especialización.

La generalización procede de observar que varios conjuntos de entidades que comparten algunas características comunes (se describen mediante los mismos atributos y participan en los mismos conjuntos de relaciones). Basada en sus similitudes, la generalización sintetiza estos conjuntos de entidades en uno solo, el conjunto de entidades de nivel más alto. La generalización se usa para resaltar las similitudes entre los conjuntos de entidades de nivel más bajo y para ocultar las diferencias; también permite economizar la representación para que los atributos compartidos no estén repetidos.

1.1.3. Herencia de atributos

Una propiedad crucial de las entidades de nivel más alto y más bajo creadas mediante especialización y generalización es la herencia de atributos. Los atributos de los conjuntos de entidades de nivel más alto se dice que son heredados por los conjuntos de entidades de nivel más bajo.

Un conjunto de entidades de nivel más bajo (o subclase) también hereda la participación en los conjuntos de relaciones en los que su entidad de nivel más alto (o superclase) participa. La herencia de atributos se aplica en todas las capas de los conjuntos de entidades de nivel más bajo.

En una jerarquía, un conjunto de entidades dado puede estar implicado como un conjunto de entidades de nivel más bajo sólo en una única relación ES. Si un conjunto de entidades es un conjunto de entidades de nivel más bajo en más

de una relación ES, entonces el conjunto de entidades tiene herencia múltiple, y la estructura resultante se denomina retículo.

1.1.4. Restricciones sobre las generalizaciones

El diseñador de la base de datos puede elegir colocar ciertas restricciones en una generalización particular. Un tipo de restricción implica determinar qué entidades pueden ser miembros de un conjunto de entidades de nivel más bajo dado. Tales relaciones de miembros pueden ser algunas de los siguientes:

- Definido por condición: En los conjuntos de entidades de nivel más bajo, la relación miembro se evalúa en función de si una entidad satisface o no una condición explícita o predicado.
- Definido por el usuario: Los conjuntos de entidades de nivel más bajo definidos por el usuario no están restringidos mediante una condición de miembro; en cambio, las entidades se asignan a un conjunto de entidades dado por el usuario de la base de datos.
- Disjunto: Una restricción sobre el carácter disjunto requiere que una entidad no pertenezca a más de un conjunto de entidades de nivel más bajo.
- Solapado: En las generalizaciones solapadas, la misma entidad puede pertenecer a más de un conjunto de entidades de nivel más bajo en una generalización simple. Se puede identificar una restricción sobre el carácter disjunto en un diagrama E-R añadiendo la palabra disjunto en el símbolo del triángulo.

Una restricción final, la restricción de completud en una generalización o especialización, especifica si un conjunto de entidades de nivel más alto debe pertenecer o no a al menos a uno de los conjuntos de entidades de nivel más bajo en una generalización/especialización. Esta restricción puede ser una de las siguientes:

- Generalización o especialización total: Cada entidad de nivel más alto debe pertenecer a un conjunto de entidades de nivel más bajo. Se puede especificar una generalización total en un diagrama E-R usando una línea doble para conectar el rectángulo que representa el conjunto de entidades de nivel más alto con el símbolo del triángulo
- Generalización o especialización parcial: Algunas entidades de nivel más alto pueden no pertenecer a algún conjunto de entidades de nivel más bajo.

1.1.5. Agregación

Una limitación del modelo E-R es que no resulta posible expresar relaciones entre relaciones. La agregación es una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto. Tal conjunto de entidades se trata de la misma forma que cualquier otro conjunto de entidades.

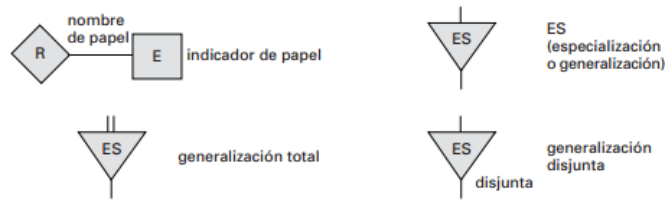


Figura 1: Elementos del diagrama ER-Extendido.

2. Base de Datos en SQL Server

Una base de datos en SQL Server es un conjunto de objetos que viven dentro de la instancia, dichos objetos ofrecen servicios que hacen posible administrar la información contenida dentro la base de datos. Algunos objetos dentro de la base de datos:

- Tablas: Almacenan datos.
- Vistas: Instrucción SQL que permite acceder a tablas.
- Procedimientos almacenados: Código ejecutado dentro de la base de datos que puede utilizarse para modificar tablas o datos dentro de la base de datos.
- Funciones: Pieza de código que realiza una tarea particular.

2.1. Tipos de base de datos

1. System database:

Bases de datos que requiere SQL Server para operar correctamente.

- master: Almacena información de la configuración SQL Server. Configuración para ingresar a SQL Server, así como información de la instancia están almacenadas en esta tabla.
- msdb: Almacena información de los jobs ejecutados, detalles de backups and historiales.
- model: Es un template para la creación de base de datos. Cada User Database es una copia de Model db.
- tempdb: Almacena los objetos temporales creados para operar correctamente las instrucciones en SQL Server. Esta base de datos es recreada cada vez que se reinicia la instancia.
- resource: Almacena los objetos del sistema.

2. User database:

Bases de datos creadas por usuarios de SQL Server, almacenan datos requeridos por los usuarios.

2.2. Logins and Users

Logins y Users son objetos de la base de datos que permiten a ciertos usuarios acceder a la instancia de SQL Server y bases de datos de la instancia. Con la creación de dichos objetos se incrementa la seguridad de nuestras bases de datos.

Un login es un objeto creado a nivel de servidor, permiten la conexión a la instancia de SQL Server. Por otro lado un User es un objeto creado a nivel de la base de datos el cual gozará de ciertos permisos sobre alguna base de datos.

Un login debe de estar mapeado a un usuario para poder conectar a una instancia de SQL Server y poder acceder a cualquier objeto en alguna base de datos. Sin este mapeo es posible realizar la conexión a la instancia pero no será posible manipular alguna base de datos.

Un usuario de la base de datos no tiene credenciales propias y necesita de un login para poder autenticarse. Por lo que no hay una contraseña asociada a un usuario.

Crear un login en SQL:

```
-- Crea un login fundamentosLogin with password 'FBD2020-1!'.
CREATE LOGIN fundamentosLogin
    WITH PASSWORD = 'FBD2020-1!';
GO
```

Crea un user en SQL:

```
-- Crea un usuario para el login fundamentosLogin
CREATE USER labFBD FOR LOGIN fundamentosLogin;
GO
```

3. Actividad

1. Se deberán aplicar los elementos del diagrama E-R extendido de todo el caso de uso seguido en el laboratorio. Deben de incluir el diagrama realizado en la practica pasada.

Deben explicar de manera muy precisa por que cambiaron algún elemento del modelo ER a ER extendido. Para el caso de generalización y especialización deben de explicar cual es la razón por la que una entidad es subclase o superclase, según sea el caso.

2. Se debe de crear un usuario 'lab_p4' para la base de de datos 'lab_db'. Dicho usuario debe tener como default la base de datos creada. Se debe

validar la siguiente salida en SQL-Transact para verificar que el usuario esté creado correctamente-

```
> sqlcmd -S localhost -U log_lab_p4 -P "miContraseña"
1> SELECT CURRENT_USER;
2> GO
```

lab_p4

```
1> SELECT DB_NAME() AS [Current Database];
2> GO
Current Database
```

lab_db

4. Entregables

Deberán agregar su archivo *.dia* en la carpeta doc. Adicional a esto, cada vez que se reúnan en equipo para discutir el diseño de la base deberán generar una bitácora sobre lo discutido en esa sesión. Deberán agregar dicha bitácora en su reporte.

El reporte debe contener las decisiones y consideraciones del diseño, las restricciones del modelo y las bitácoras.

También deberán agregar un script con las instrucciones necesarias para crear el usuario de la actividad 2 así como adjuntar una imagen al reporte con la evidencia para verificar que el usuario fue creado correctamente.

La entrega deberá ser el día martes 17 de Septiembre de 2019.