

Diseño de BD relacionales

Introducción

Fases del diseño

El diseño de una Base de Datos (BD) consta de **tres fases**: diseño conceptual, diseño lógico y diseño físico.

En la primera se crea un modelo Entidad-Relación (E-R) para representar la información que se va a almacenar. A continuación, se realiza el diseño lógico, que consiste en pasar el modelo E-R a tablas, normalizándolas.

Por último, se lleva a cabo el diseño físico, en el que se genera el esquema físico de la BD sobre el Sistema Gestor de Bases de Datos (SGDB) escogido, se definen las tablas y campos, se imponen las restricciones de integridad, y se definen los índices.

Diseño conceptual

Conceptos:

- Entidades
- Relaciones. Grado de una relación
- Atributos (multivalorados, compuestos)

Diagramas entidad-relación

- Elección de los tipos de entidad y sus atributos
- Elección de tipos de relación

Diseño lógico

Del modelo E/R al modelo lógico (esquema E/R a esquema relacional)

Transformar E/R a tablas

Para cada entidad se crea una relación con el mismo nombre y conjunto de atributos

Transform atributos en campos de tablas

Para cada relación se crea una relación que tiene atributos de las entidades relacionadas

Relaciones tienen clave compuesta de las entidades que relaciona

Claves

Restricciones de integridad

- Integridad referencial:
 - No puede aparecer un DNI matrícula que no esté en DNI de alumnos
- Participación total

- Todos los valores de DNI de Alumnos debe aparecer en DNI de matrícula

Normalización

Formas normales

- Primera FN
- Segunda FN
- Tercera FN
- Forma normal de Boyce-Codd
- Cuarta FN
- Quinta FN

¿Es necesario llegar a la quinta FN?

Ejemplo de normalización

- Primera
- Segunda
- Tercera

Diseño físico. Optimización

- Índices
 - Elección de índices
 - Índices en SQL
- Vistas
- Estadísticas
- Usuarios y permisos

Bibliografía

Forth y Silberschatdz "Fundamentos de bases de datos" (McGraw-hill)

Construir una BD desde el modelo de datos

Primer paso

crear una relación para cada entidad fuerte en el modelo de datos. Cada atributo del tipo de entidad en el modelo se convertirá en un nombre de columna en la relación. En este momento uno debe elegir un atributo, o conjunto de atributos, llamado una clave primaria, que identificará de manera única cada fila de la relación. La clave ideal es corta, numérica y nunca cambia. Si no hay un buen atributo clave entre los atributos de la tabla. Una opción es concatenar los valores de varios campos para lograr un identificador que será único para cada fila. Si este enfoque conduce a claves largas, alfanuméricas, puede ser mejor usar una clave sustituta. Una clave sustituta es simplemente un número, generado por el DBMS, que se asigna a cada tupla.

Segundo paso

Crear una relación para cada tipo de entidad débil y dependiente de ID. Cada atributo del tipo de entidad en el modelo de datos se convierte en una columna en la nueva relación. Se debe agregar una columna a la

relación débil o dependiente de ID que contenga la clave externa de la tupla de entidad fuerte a la que está relacionada. Clave externa Columna en una relación, establece una relación con los datos en otra relación.

Además de los atributos de la computadora del estudiante, tales como marca y número de serie, la relación StudentComputer tendrá una columna que identifica al estudiante que posee la computadora. Si SSN es la clave de la relación Estudiante, la clave extranjera en la relación StudentComputer contendrá los valores de los números de seguro social de los estudiantes. No es necesario que los nombres de columna en las dos relaciones sean los mismos. Por lo tanto, aunque la columna clave de la | Relación de estudiante se denomina "SSN", la columna de clave externa en StudentComputer se podría llamar "StudentSSN". La nueva relación creada para la entidad débil también debe tener una clave primaria propia. Si la entidad débil es dependiente del ID de la entidad fuerte, entonces haga que la clave de la relación dependiente del ID sea una combinación del campo de clave externa y uno o más otros atributos de la relación dependiente del ID. Otra aplicación de entidades dependientes de ID es el modelado de atributos multivalorados. Por ejemplo, es posible que desee proporcionar múltiples direcciones para cada estudiante; Muchos tendrán una dirección durante el año académico y otra durante el verano, por ejemplo. En tal caso, modelar una entidad dependiente de ID denominada "Dirección", y crear una relación con atributos tales como "Calle", "Ciudad", "Estado", etc., así como un atributo de clave externa que contenga valores De la clave primaria para la relación Estudiantil.