Guía Instructiva: Creación de un Modelo de Datos Relacional (Conceptual y Lógico)

Crear un modelo de datos relacional es un paso fundamental en el diseño de cualquier sistema de información. Nos permite entender y organizar la información de manera estructurada antes de pensar en cómo se almacenará físicamente. Esta guía te ayudará a transitar desde la idea de un sistema hasta un diseño de base de datos sólido.

1. Modelo de Datos Conceptual: Entendiendo el Negocio

El modelo conceptual es el primer paso. Su objetivo es identificar las **entidades** (las "cosas" o sujetos importantes para el sistema) y las **relaciones** entre ellas, sin preocuparse aún por los detalles técnicos de la base de datos. Pensá en este paso como un "dibujo" de alto nivel de la información.

1.1. Identificación de Entidades

Una entidad representa un objeto, un evento o un concepto del mundo real sobre el cual queremos almacenar información.

- **Pregunta clave:** ¿Sobre qué "cosas" o "sujetos" necesita mi sistema guardar datos?
- **Ejemplos:**
    - **Usuarios:** Clientes, administradores, empleados, etc.
    - **Productos:** Artículos que se venden.
    - **Pedidos:** Transacciones de compra.
    - **Publicaciones:** Artículos de un blog, posts en una red social.
    - **Eventos:** Actividades, conferencias.
- **Consejo:** Pensá en sustantivos. Evitá los verbos, ya que suelen describir acciones o relaciones.

1.2. Definición de Atributos para Cada Entidad

Los atributos son las características o propiedades que describen una entidad.

- **Pregunta clave:** ¿Qué información específica necesito almacenar sobre cada entidad?
- **Ejemplos para la entidad "Usuario":**
    - `nombre`
    - `apellido`
    - `email` (idealmente único)
    - `fecha_registro`
    - `contraseña`
    - `telefono`
- **Consejo:** Evitá atributos que puedan ser calculados a partir de otros (ej. `edad` si tenés `fecha_nacimiento`). Pensá en qué datos son fundamentales y atómicos (no se pueden dividir más).

1.3. Identificación de Claves Primarias (PK)

Una **clave primaria (PK)** es uno o más atributos que identifican de forma única cada instancia (fila) de una entidad. Es como el DNI de cada registro.

- **Características:**
  - **Única:** No se repite para dos registros diferentes.
  - **No nula:** Siempre debe tener un valor.
  - **Estable:** Su valor no debería cambiar a lo largo del tiempo.
- **Ejemplos:**
  - `id_usuario` para la entidad Usuarios.
  - `dni` para la entidad Personas (si es único y no cambia).
- **Consejo:** A menudo, se utiliza un `ID` numérico o alfanumérico generado automáticamente (`UUID` o `ObjectId` en NoSQL como MongoDB) como clave primaria.

1.4. Definición de Relaciones entre Entidades

Las relaciones describen cómo las entidades interactúan o se conectan entre sí.

- **Pregunta clave:** ¿Cómo se vinculan mis entidades entre sí?
- **Tipos de Relaciones (Cardinalidad):**
  - **Uno a Uno (1:1):** Una instancia de la Entidad A se relaciona con una única instancia de la Entidad B, y viceversa.
    - *Ejemplo:* Un `Usuario` tiene un único `PerfilDeContacto` y viceversa.
  - **Uno a Muchos (1:N):** Una instancia de la Entidad A puede relacionarse con múltiples instancias de la Entidad B, pero una instancia de la Entidad B solo se relaciona con una única instancia de la Entidad A.
    - *Ejemplo:* Un `Usuario` puede tener muchas `Publicaciones`, pero una `Publicación` pertenece a un único `Usuario`.
  - **Muchos a Muchos (N:M):** Una instancia de la Entidad A puede relacionarse con múltiples instancias de la Entidad B, y viceversa.
    - *Ejemplo:* Un `Estudiante` puede cursar muchas `Materias`, y una `Materia` puede ser cursada por muchos `Estudiantes`.
- **Cómo representarlas:** Dibujá líneas entre las entidades y etiquetá el tipo de relación con su cardinalidad.

1.5. Diagrama Entidad-Relación (ERD)

Un ERD es la representación visual de tu modelo conceptual. Usá símbolos estándar para entidades, atributos y relaciones. Podés usar herramientas como `draw.io`, `Lucidchart`, `dbdiagram.io` o incluso dibujar a mano.

---

2. Modelo de Datos Lógico: Preparando la Base de Datos

El modelo lógico toma el modelo conceptual y lo traduce a estructuras que pueden ser implementadas en una base de datos relacional (tablas, columnas, claves foráneas), aunque el objetivo final sea MongoDB. Es el paso intermedio que facilita la transición.

2.1. Transformación de Entidades a Tablas

Cada entidad de tu modelo conceptual se convierte en una **tabla** en el modelo lógico.

- **Nombre de la Tabla:** Generalmente, el nombre de la entidad en plural (ej. `Usuarios`, `Productos`).

2.2. Transformación de Atributos a Columnas

Cada atributo de una entidad se convierte en una **columna** dentro de la tabla correspondiente.

- **Nombre de la Columna:** Un nombre descriptivo (ej. `email`, `fecha_registro`).
- **Tipo de Dato:** Aunque no es el enfoque principal para MongoDB, es útil pensar en un tipo de dato lógico (ej. `VARCHAR(255)` para texto, `INT` para números enteros, `BOOLEAN` para verdadero/falso, `DATE` o `DATETIME` para fechas).

2.3. Implementación de Claves Primarias (PK)

La clave primaria identificada en el modelo conceptual se convierte en la **columna de clave primaria** de la tabla.

- **Ejemplo:** La tabla `Usuarios` tendrá una columna `id_usuario` marcada como PK.

2.4. Implementación de Relaciones con Claves Foráneas (FK)

Las relaciones entre entidades se implementan usando **claves foráneas (FK)**. Una FK es una columna en una tabla que hace referencia a la clave primaria de otra tabla.

- **Reglas de Transformación:**
    - **1:1 (Uno a Uno):** La clave primaria de una tabla se convierte en clave foránea en la otra tabla (cualquiera de las dos, por lo general la que tenga más sentido lógico).
        - *Ejemplo:* `PerfilDeContacto` tendrá `id_usuario` (FK) que referencia a `Usuarios.id_usuario`.
    - **1:N (Uno a Muchos):** La clave primaria de la tabla "uno" se convierte en clave foránea en la tabla "muchos".
        - *Ejemplo:* En la tabla `Publicaciones`, se agrega una columna `id_usuario_creador` (FK) que referencia a `Usuarios.id_usuario`.
    - **N:M (Muchos a Muchos):** Se crea una **nueva tabla intermedia** (también llamada tabla de unión o pivote) para resolver la relación. Esta nueva tabla tendrá como claves foráneas las claves primarias de ambas tablas que se relacionan.
        - *Ejemplo:* Para `Estudiantes` y `Materias`, creás una tabla `Estudiantes_Materias` con `id_estudiante` (FK de Estudiantes) y `id_materia` (FK de Materias), y ambas FK juntas forman la clave primaria compuesta de la tabla intermedia.

2.5. Diagrama Lógico de la Base de Datos

Actualizá tu ERD para que muestre claramente las tablas, columnas (con tipos de datos lógicos) y cómo las PK y FK establecen las relaciones. Herramientas como `dbdiagram.io`

son excelentes para esto, ya que te permiten definir las tablas y relaciones y te generan el diagrama automáticamente.

Consejos Adicionales

- **Normalización:** Es un proceso para organizar las columnas y tablas para minimizar la redundancia de datos y mejorar la integridad de los mismos. Aunque para MongoDB a veces se "desnormaliza" para optimizar lecturas, entender la normalización es clave para un buen diseño inicial. Para este TP, apuntá a un diseño normalizado (sin redundancia).
- **Claridad:** Mantené los nombres de tus tablas, columnas y relaciones claros y consistentes.
- **Iteración:** El diseño de bases de datos es un proceso iterativo. Es normal que encuentres la necesidad de ajustar tu modelo a medida que avanzás.