Elementos de una base de datos

¡Comencemos!

¿Qué aprenderás?

- Interpretar el concepto de tabla en una base de datos.
- Reconocer cómo se relacionan las tablas mediante claves primarias y foráneas.
- Clasificar los tipos de datos que se pueden utilizar en una base de datos.

Introducción

En este capítulo aprenderás la diferencia entre una base de datos relacional y una base de datos no relacional, para que conozcas el contraste que hay entre estos dos mundos alternos y refuerces tu capacidad de análisis en la toma de decisiones, al crear un software que necesite persistencia de datos. Continuando con el ejemplo del capítulo anterior, empezarás a crear las relaciones entre las tablas por medio de claves primarias y claves foráneas, las cuales sirven para definir las dependencias de las entidades.

Finalmente, aprenderás los diferentes tipos de datos que se almacenan en columnas dentro de las tablas, logrando restringir el ingreso de datos incoherentes a una entidad de contexto definido dentro de las bases de datos.

Bases de datos relacionales versus bases de datos no relacionales

Existen dos tipos de bases de datos: las Relacionales y las No Relacionales.

- Las bases de datos relacionales son aquellas compuestas por varias tablas donde se almacena la información, y posteriormente se relacionan entre sí.

- Las bases de datos No Relacionales son aquellas que siguen esquemas más flexibles de organización, donde no necesariamente todas las entradas tienen la misma estructura.

Un aspecto relevante a tomar en cuenta, es que para implementar bases de datos relacionales o no relaciones, es necesario tener conocimiento previo sobre qué es lo que vamos a almacenar, teniendo en cuenta los siguientes puntos clave:

Relacional	No Relacional
Cuando el volumen de datos no crece o lo hace gradualmente.	Cuando el volumen de datos crece muy rápidamente.
Cuando las necesidades de proceso se pueden asumir en un solo servidor o en N servidores definidos previamente.	Cuando las necesidades del proceso no se pueden prever.
Cuando no existen peaks de uso o son esporádicos.	Cuando existen peaks de uso en múltiples ocasiones.
Cuando requerimos mantener la integridad referencial.	Cuando la información no requiere mantener relación entre los registros.
Estructura de datos mayormente estática.	Estructura de datos variable.

Tabla 4. Relacional v/s No Relacional.
Fuente: Desafío Latam.

Como podemos observar, existen modelos más aptos dependiendo de la naturaleza del negocio que estamos abordando. En esta unidad sólo veremos cómo operar con bases de datos de tipo relacional.

Ejercicio guiado: directorio telefónico

Crear un registro telefónico donde alojaremos el nombre, apellido, número telefónico, dirección y edad de una serie de individuos. Resulta que el registro en sí será la tabla **Directorio_telefonico**, donde tendremos todos los campos mencionados anteriormente.

Para empezar a crear nuestra base de datos utilizaremos tablas, donde alojaremos esta información.

Tablas

Una base de datos se compone de múltiples tablas. Cada una de éstas presentarán dos dimensiones:

- Filas, que representan a los registros en la tabla.
- Columnas, que van a representar los atributos ingresados en cada registro, definiendo el tipo de dato a ingresar.

Claves primarias y foráneas

De manera adicional a las filas y columnas, las tablas también cuentan con claves primarias y foráneas. Estas buscan generar identificadores para cada registro de estas tablas mediante algún valor específico de una columna o atributo.

Clave primaria (primary key)

Cuando hacemos referencia a una columna dentro de su tabla de origen, hablaremos de una clave primaria. Esta clave siempre será de carácter único.

Clave foránea (foreign key)

Cuando hacemos referencia a una columna identificadora en otra tabla a la cual hacemos referencia, hablamos de una clave foránea.

Veamos esto con el ejemplo

Supongamos que en base a nuestra tabla `Directorio_telefonico`, deseamos incorporar información de otra tabla llamada `Agenda` que tiene las columnas `nick` y `numero_telefonico`. Ante esta situación, vamos a identificar que la columna `numero_telefonico` en la tabla `Directorio_telefonico` será la clave primaria y la columna `numero_telefonico` en la tabla Agenda corresponderá a la clave foránea. Este comportamiento es posible dado que el número registrado será congruente en ambas tablas, en la siguiente imagen te muestro una referencia de esto.

Nota: Se recomienda siempre crear los nombres de los campos (columnas) sin tildes ni caracteres que contengan símbolos como "ñ".

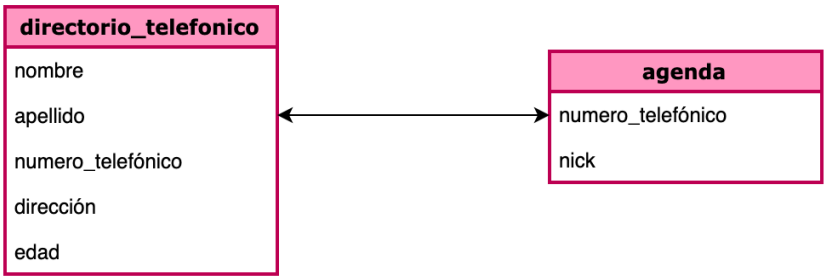

Imagen 12. Modelo relacional.
Fuente: Desafío Latam.

Tipos de datos

Una de las principales características del trabajo con bases de datos, es la necesidad de declarar los distintos tipos de datos existentes en cada campo a completar. Estos aplican restricciones sobre lo que pueden ingresar a los registros. No podemos ingresar caracteres cuando piden un número, ni sobrepasar el límite de caracteres posibles.

Los tipos de datos más comunes son:

- `INT`: Números enteros de 4 bytes que pueden tomar valor desde -2147483648 hasta +2147483647.
- `SMALLINT`: Números enteros de 2 bytes que pueden tomar valor desde -32768 hasta +32767.
- BIGINT: Números enteros de 8 bytes que pueden tomar valor desde -9223372036854775808 hasta 9223372036854775807.
- `FLOAT`: Números decimales de 32 bit de precisión.

- `DOUBLE`: Números decimales de 64 bit de precisión con hasta 15 dígitos decimales.
- `CHAR`: Cadena de hasta 255 caracteres de longitud fija.
- `VARCHAR`: Cadena de hasta 65.535 caracteres de longitud variable. A diferencia de CHAR, si no se ocupa toda la memoria, esta queda libre. CHAR ocupará toda la memoria solicitada.
  Para CHAR Y VARCHAR se puede definir un límite máximo de caracteres como CHAR(N) Y VARCHAR(N), si se trata de almacenar un valor que sobrepase el límite, PostgreSQL entregará un error.
- `DATE`: Almacena fecha en formato aaaa-mm-dd. <año>-<mes>-<dia>.
- `TIME`: Almacena el tiempo en horario desde 00:00:00 hasta 24:00:00.
- `TIMESTAMP`: Almacena fecha y hora juntos: yyyy-mm-dd hh:mm:ss.
- `BOOLEAN`: Tiene 3 valores posibles: Verdadero, Falso o NULL.
  Estos pueden representarse para el caso:
  - Verdadero: 1, yes, t o true.
  - Falso: 0, no, false o f.