

Elementos de una base de datos

Elementos de una base de datos	1
¿Qué aprenderás?	2
Introducción	2
Bases de datos relacionales versus bases de datos no relacionales	3
Tablas	4
Claves primarias y foráneas	4
Ejemplo práctico: Directorio telefónico	4
Tipos de datos	6



¡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 tablas y posteriormente sus relaciones por medio de claves primarias y claves foráneas, las cuales sirven para definir las dependencias de las entidades; pero como podremos imaginar, lo veremos con más detalle durante el recorrido del presente capítulo.

Adicionalmente, aprenderemos 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.

¡Vamos con todo!



Bases de datos relacionales versus bases de datos no relacionales

Podemos clasificar dos tipos de bases de datos: las Relacionales y las No Relacionales.

- Las bases de datos relacionales, son aquellas compuestas por una o 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 1. Relacional v/s no relacional.

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

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.

Ejemplo práctico: Directorio telefónico

Es normal querer innovar con aplicaciones que no existan o que agreguen un gran valor. Es por eso que, a pesar que tenemos en nuestros móviles un directorio lo suficientemente sofisticado, podemos aplicar ejemplos que nos permitan madurar, para luego desplazar a Zuckerberg; pero apliquemos algunos análisis, que nos permitirán saber, cómo almacenar y administrar la información de un directorio telefónico básico.

Supongamos que deseamos crear un registro telefónico donde alojaremos:

- Nombre
- Apellido

- Número
- Telefónico
- Dirección
- 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.

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.

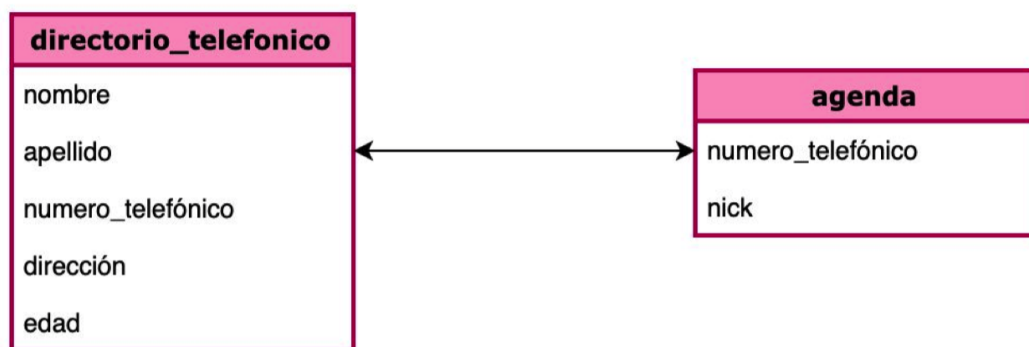


Imagen 1. Modelo relacional.

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.

La siguiente lista indica los tipos de datos más comunes, es decir aquellos que utilizaremos en el programa, para más información respecto a los tipos de datos de Oracle, visitar el siguiente [enlace](#).

- **VARCHAR2.**
 - Para guardar los tipo String.
 - Debemos indicar el largo, ejemplo: nombre VARCHAR2(200).
 - Cadena de caracteres de longitud variable.
 - Entre 1 y 4000 bytes como máximo. El tamaño del campo dependerá del valor que contenga, es de longitud variable.
- **NUMBER.**
 - Almacena números fijos y en punto flotante.
 - Se admiten hasta 38 dígitos de precisión y son portables a cualquier entre los diversos sistemas en que funcione Oracle.
 - Para declarar un tipo de datos NUMBER en un CREATE ó UPDATE es suficiente con: nombre_columna NUMBER.
 - **Ejemplo:** edad Number.
 - Opcionalmente se le puede indicar la precisión (número total de dígitos) y la escala (número de dígitos a la derecha de la coma, decimales, los cogerá de la precisión indicada): nombre_columna NUMBER (precisión, escala).
 - **Ejemplo:** área Number(4, 2), indica que el número total de dígitos es 4, incluyendo los dos decimales, es decir que, admite dos dígitos enteros y dos dígitos decimales.
 - Serían admitidos: 2, 2.2 o 22.22
 - No serían admitidos: 222, 222.2 o 222.22
 - Si no se indica la precisión se tomará en función del número a guardar, si no se indica la escala se tomará escala cero.

- Para no indicar la precisión y sí la escala podemos utilizar: nombre_columna NUMBER (*, escala).
 - Para introducir números que no estén en el formato estándar de Oracle se puede utilizar la función TO_NUMBER.
- **DATE**
 - Almacena un punto en el tiempo (fecha y hora).
 - El tipo de datos DATE almacena el año (incluyendo el siglo), el mes, el día, las horas, los minutos y los segundos (después de medianoche).
 - Oracle utiliza su propio formato interno para almacenar fechas.
 - Los tipos de datos DATE se almacenan en campos de longitud fija de siete octetos cada uno, correspondiendo al siglo, año, mes, día, hora, minuto, y al segundo.
 - Para entrada/salida de fechas, Oracle utiliza por defecto el formato DD-MM-AA. Para cambiar este formato de fecha por defecto se utiliza el parámetro NLS_DATE_FORMAT.
 - Para insertar fechas que no estén en el mismo formato de fecha estándar de Oracle, se puede utilizar la función TO_DATE con una máscara del formato: TO_DATE (el "13 de noviembre de 1992", "DD del MES, YYYY").



En resumen, los tipos de datos con los que quizás nos enfrentemos más comúnmente incluso en la etapa laboral serían los siguientes (los últimos dos muy poco probable):

- **VARCHAR2:** Cadenas de texto no muy grandes.
- **DATE:** Fechas, horas.
- **NUMBER:** Números.
- **BLOB:** Ficheros de tipo word, excel, access, video, sonido, imágenes, etc.
- **CLOB:** Cadenas de texto muy grandes.