



Asignatura:

Programación Orientada A Objetos

Taller Bases de Datos – NetBeans

Alumno

Michael Daniel Murillo López

ID:534830

Docente

Alonso Guevara Pérez

NRC:6293

Taller Bases de Datos – NetBeans

- Que es una base de Datos
- Diccionario de la Base de datos
- Tipos de Datos en MySQL
- Que es y para qué sirve SQL
- Crear una base de datos en MySQL por consola que tenga los siguientes campos:
 - a) Nombre apellidos persona.
 - b) Edad
 - c) Teléfono
 - d) Sexo
 - e) Dirección Física
 - f) Correo Electrónico
 - g) NUI Numero Único de Identificación
 - h) Nacionalidad
 - i) Lugar de Nacimiento
- Conectar la Base de Datos de MySQL a Java utilizando NetBeans
- Crear consultas desde NetBeans En abierta comunicación.

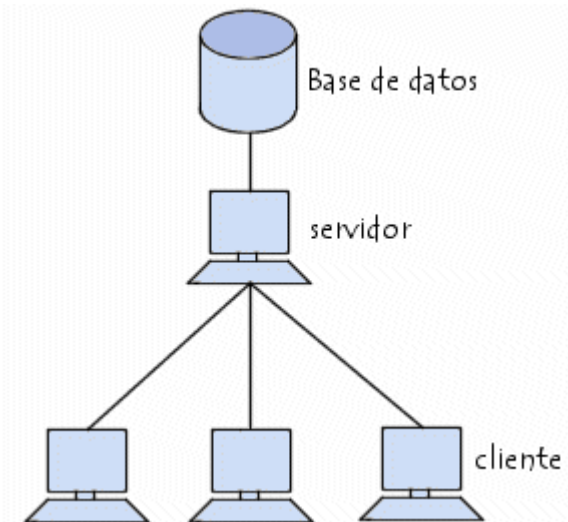
El Ejercicio hay mencionado se encuentra en el siguiente Link.

<https://github.com/michaeldanielm/Registro-MySQL>

Que es una base de Datos

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono. A veces se utiliza DB, de database en inglés, para referirse a las bases de datos.



Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red, ya que se debe poder compartir esta información. Generalmente se habla de un "Sistema de información" para designar a la estructura global que incluye todos los mecanismos para compartir datos.

Diccionario de la Base de Datos

Las vistas de base de datos son tablas de bases de datos virtuales que muestran los datos seleccionados de las tablas de bases de datos. Puede utilizar las vistas de base de datos para extraer los datos seleccionados almacenados en la base de datos de License Metric Tool.

Parte de la información que se describe en las vistas de base de datos, como la información sobre los grupos de exploración, también está disponible en la interfaz de usuario Web. Otro tipo de información, como los datos de inventario de hardware, solamente está disponible en las vistas de base de datos. El acceso a las vistas es de sólo lectura.

Tipos de Datos en MySQL

Tipos de dato en una base de datos MySQL

Al crear una tabla la elección correcta de un formato de dato para cada columna de la tabla hará que nuestra BBDD tenga un rendimiento óptimo a medio largo plazo.

Repasando la documentación oficial podríamos dividir en 3 grandes grupos estos datos:

- Numéricos
- Fecha
- String

Tipos de dato numéricos

Listado de cada uno de los tipos de dato numéricos en MySQL, su ocupación en disco y valores.

- ❖ INT (INTEGER): Ocupación de 4 bytes con valores entre -2147483648 y 2147483647 o entre 0 y 4294967295.
- ❖ SMALLINT: Ocupación de 2 bytes con valores entre -32768 y 32767 o entre 0 y 65535.
- ❖ TINYINT: Ocupación de 1 bytes con valores entre -128 y 127 o entre 0 y 255.
- ❖ MEDIUMINT: Ocupación de 3 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215.
- ❖ BIGINT: Ocupación de 8 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215.
- ❖ DECIMAL (NUMERIC): Almacena los números de coma flotante como cadenas o string.
- ❖ FLOAT (m,d): Almacena números de coma flotante, donde 'm' es el número de dígitos de la parte entera y 'd' el número de decimales.
- ❖ DOUBLE (REAL): Almacena número de coma flotante con precisión doble. Igual que FLOAT, la diferencia es el rango de valores posibles.
- ❖ BIT (BOOL, BOOLEAN): Número entero con valor 0 o 1.

Tipos de dato con formato fecha

Listado de cada uno de los tipos de dato con formato fecha en MySQL, su ocupación en disco y valores.

- ❖ DATE: Válido para almacenar una fecha con año, mes y día, su rango oscila entre '1000-01-01' y '9999-12-31'.
- ❖ DATETIME: Almacena una fecha (año-mes-día) y una hora (horas-minutos-segundos), su rango oscila entre '1000-01-01 00:00:00' y '9999-12-31 23:59:59'.
- ❖ TIME: Válido para almacenar una hora (horas-minutos-segundos). Su rango de horas oscila entre -838-59-59 y 838-59-59. El formato almacenado es 'HH:MM: SS'.
- ❖ TIMESTAMP: Almacena una fecha y hora UTC. El rango de valores oscila entre '1970-01-01 00:00:01' y '2038-01-19 03:14:07'.
- ❖ YEAR: Almacena un año dado con 2 o 4 dígitos de longitud, por defecto son 4. El rango de valores oscila entre 1901 y 2155 con 4 dígitos. Mientras que con 2 dígitos el rango es desde 1970 a 2069 (70-69).

Diferentes tipos de dato con formato string

Listado de cada uno de los tipos de dato con formato string en MySQL, su ocupación en disco y valores.

- ❖ CHAR: Ocupación fija cuya longitud comprende de 1 a 255 caracteres.
- ❖ VARCHAR: Ocupación variable cuya longitud comprende de 1 a 255 caracteres.
- ❖ TINYBLOB: Una longitud máxima de 255 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- ❖ BLOB: Una longitud máxima de 65.535 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- ❖ MEDIUMBLOB: Una longitud máxima de 16.777.215 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- ❖ LONGBLOB: Una longitud máxima de 4.294.967.298 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- ❖ SET: Almacena 0, uno o varios valores una lista con un máximo de 64 posibles valores.
- ❖ ENUM: Igual que SET, pero solo puede almacenar un valor.
- ❖ TINYTEXT: Una longitud máxima de 255 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.
- ❖ TEXT: Una longitud máxima de 65.535 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.
- ❖ MEDIUMTEXT: Una longitud máxima de 16.777.215 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.

- ❖ LONGTEXT: Una longitud máxima de 4.294.967.298 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.

Que es y para qué sirve SQL

La Sigla que se conoce como SQL corresponde a la expresión inglesa Structured Query Language (entendida en español como Lenguaje de Consulta Estructurado), la cual identifica a un tipo de lenguaje vinculado con la gestión de bases de datos de carácter relacional que permite la especificación de distintas clases de operaciones entre éstas. Gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla.

El científico Edgar Frank Codd (1923–2003) fue quien propuso un modelo relacional para las bases de datos y creó un sublenguaje para acceder a los datos a partir del cálculo de predicados. En base al trabajo de Codd, IBM (International Business Machines) definió el lenguaje conocido como Structured English Query Language (SEQUEL). El SEQUEL se considera el antecesor de SQL, un lenguaje de cuarta generación que se estandarizó en 1986. La versión más primitiva de SQL, por lo tanto, fue la que se bautizó como SQL-86 (también conocida como SQL1).

En esencia, el SQL es un lenguaje declarativo de alto nivel ya que, al manejar conjuntos de registros y no registros individuales, ofrece una elevada productividad en la codificación y en la orientación a objetos. Una sentencia de SQL puede resultar equivalente a más de un programa que emplee un lenguaje de bajo nivel.

Una base de datos, dicen los expertos, implica la coexistencia de múltiples tipos de lenguajes. El denominado Data Definición Language (también conocido como DDL) es aquél que permite modificar la estructura de los objetos contemplados por la base de datos por medio de cuatro operaciones básicas. SQL, por su parte, es un lenguaje que permite manipular datos (Data Manipulation Language o DML) que contribuye a la gestión de las bases de datos a través de consultas.

¿Cómo construir una base de datos eficiente?

Toda empresa que apunte a un futuro brillante, con posibilidades de crecimiento y expansión, debe contar con una base de datos, que será diferente en cada caso, ajustándose a las necesidades particulares de cada tipo de negocio, pero que deberá ser confeccionada cuidadosamente, con una estructura sólida y configurable, abierta a potenciales modificaciones sin que esto amenace su integridad.

Uno de los puntos básicos a la hora de construir una base de datos es la indexación. Para entender este concepto, veamos brevemente un ejemplo práctico de base: supongamos que una compañía desea almacenar la información personal de sus clientes y hacer un seguimiento de sus transacciones; para ello, una posibilidad consistiría en tener una tabla para sus datos (nombre, apellido, dirección de e-mail, etcétera), otra para la descripción de los productos (nombre del artículo, precio, detalles) y una para las ventas. Antes de pasar a detallar qué campos podrían estar presentes en esta última tabla, cabe mencionar que en las restantes falta un elemento indispensable para una buena organización: una clave única de identificación.

Generalmente llamada ID, suele ser un número entero (sin decimales) y positivo que la base de datos asigna automáticamente a cada nuevo registro (en este caso, cada nuevo cliente o producto) y que nunca se repite, de modo que lo identifique desde su nacimiento (momento de creación) hasta su muerte (cuando se elimine). De esta forma, si tomamos por ejemplo el registro “103 Pablo Bernal pbernal@proveedor.com”, notamos que su ID es 103. ¿Cuál es su utilidad? En pocas palabras, buscar un cliente cuyo nombre sea n, su apellido, a, y su e-mail, e, toma mucho más tiempo que pedir a la base que nos devuelva todos los datos del cliente con ID “103”. Si bien es probable que en la primera operación especifiquemos toda su información, una vez que el programa lo encuentre, podremos valernos de este número para el resto de las consultas.

Retomando el ejemplo, dado que las tablas de clientes y productos tendrían su ID, relacionarlas con la de ventas resulta muy sencillo; sus campos podrían ser: id de transacción, id de cliente, id de producto, fecha, observaciones. Sin entrar en tecnicismos, es claro que cada venta contiene mucha más información de la que se aprecia a simple vista, ya que, por ejemplo, el id de un cliente nos sirve para acceder a todos sus datos en la tabla correspondiente. En la puesta en práctica, sobra decir que la complejidad puede ser muchísimo mayor, pero es importante comenzar por lo básico para entender la importancia de relaciones sólidas y eficientes.

SQL es un lenguaje completamente normalizado que facilita el trabajo con cualquier tipo de lenguaje a la par con cualquier tipo de Base de Datos, sin embargo, esto no es equivalente a decir que es igual en todos los MBD, estos implementan diferentes funciones de acuerdo a la manera como mas favorezca al MBD, estas funciones no siempre funcionan en otros.

Referencias

Masadelante.com

<http://www.masadelante.com/faqs/base-de-datos>

Ccmbenchmark.com

<http://es.ccm.net/contents/66-introduccion-a-las-bases-de-datos>

IBM Knowledge Center

https://www.ibm.com/support/knowledgecenter/es/SS8JFY_7.5.0/com.ibm.lmt75.doc/com.ibm.license.mgmt.reference.doc/c_dboverview.html

Aner barrena.com 19 Agosto, 2015

<http://www.anerbarrena.com/tipos-dato-mysql-5024/>

Definicion.de

<http://definicion.de/sql/>