

La primera base de datos

3° Ciclo y Polimodal



Didáctica de la actividad

Información de la actividad

Recursos

El objetivo de esta actividad es crear una base de datos con el monitor MySQL, uno de los programas abiertos más populares para armar bases de datos. La propuesta es combinar el MySQL con el DBDesigner, una herramienta de diseño y edición de bases de datos.

Para esto, vamos a trabajar en varias etapas.

Etapas 1: Crear una base de datos vacía en nuestro servidor local de MySQL

En esta etapa se utilizará el monitor, que es una interfaz de línea de comandos. ¡A no asustarse! Los pasos son simples y los enumeramos aquí.

1. Iniciar el monitor de MySQL a través de los siguientes comandos:

Menú de Inicio -> Programas -> MySQL -> MySQL Server 4.1 -> MySQL 4.1 Command Line Client

2. Escribir la clave del superusuario (llamado root).
3. Escribir la siguiente orden, que creará la base de datos:

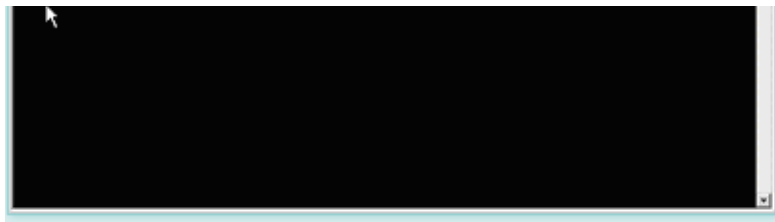
`create database primera;`

4. Salir del monitor con la orden:
`quit;`

No olvidar el punto y coma al final de cada orden.

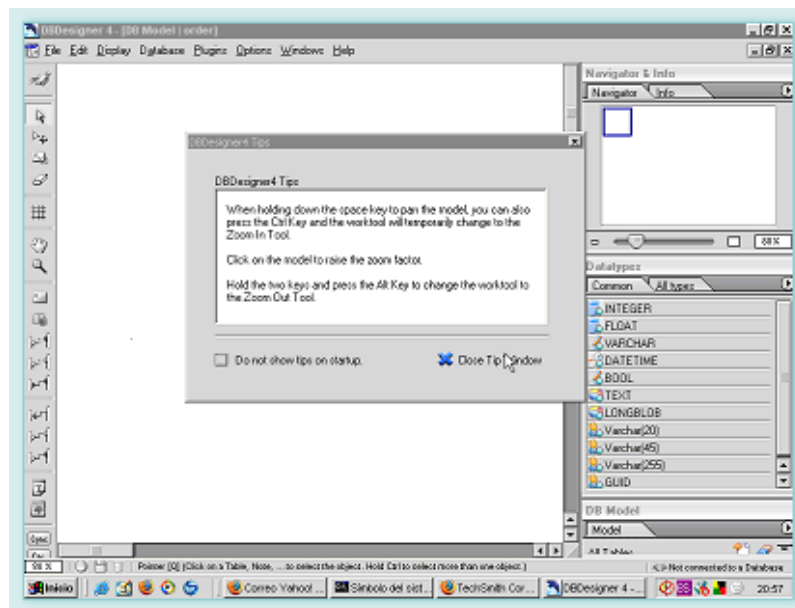
Así, hemos creado la base de datos de nombre: **primera**. Vemos en el monitor MySQL, la siguiente pantalla:

```
MySQL Command Line Client
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 4.1.9-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> create database primera;
```

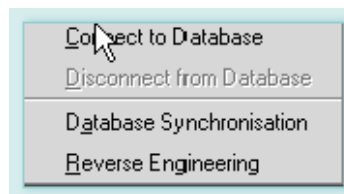


Etapas 2: Utilizar el DBDesigner

1. Seleccionar el ícono que quedó después de la instalación en el escritorio para iniciar el DBDesigner. El DBDesigner es un "escritorio" para diseñar las tablas de las bases de datos.
2. Se llegará a una pantalla de diseño en blanco. Puede aparecer una sugerencia o "Tip" que puede desactivarse haciendo clic en "Do not show tips on startup".



3. Conectarse a la base de datos local MySQL desde el menú, tal como vemos en la imagen.
Database -> Connect Database

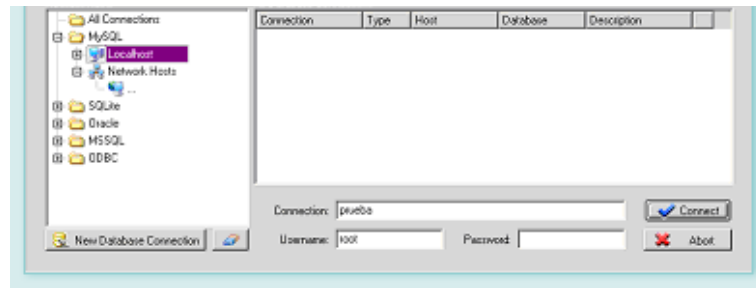


El DBDesigner no crea las bases de datos dentro del servidor MySQL, sólo crea elementos dentro de ella. Por esta razón para llevar a cabo esta etapa deben estar cumplidas las tareas de la etapa anterior:

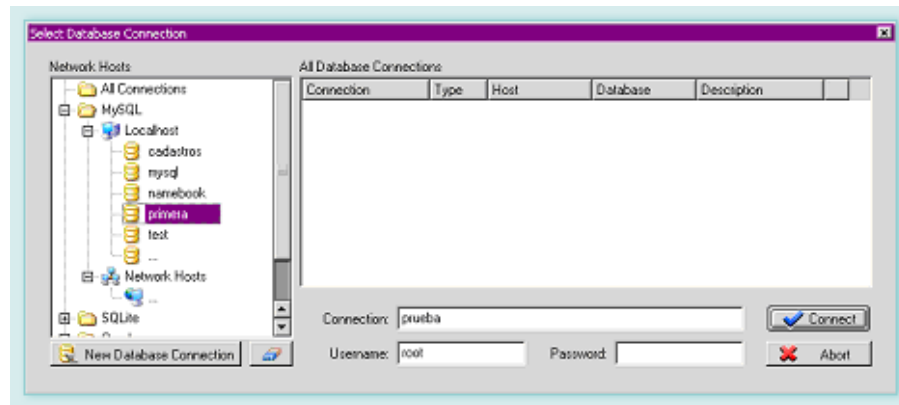
- Debe estar funcionando el servidor de MySQL en nuestra máquina.
- Debe existir una base de datos -como por ejemplo, "primera". La base de datos se llenará luego con las tablas que diseñemos con el DBDesigner.

4. Buscar el servidor -que en este caso es la propia máquina- con el nombre localhost, que significa "anfitrión local" en inglés.





5. Al abrir la carpeta **localhost**, se observarán las bases de datos que estén definidas. Seleccionar "primera".



6. Llenar los campos "Conection", "Username" y "Password" para crear una conexión a la base de datos "primera" y aceptar presionando el tilde color azul. Para cancelar, presionar la cruz roja.

Etap 3: Construir las tablas de la base de datos "primera"

¿Qué son las tablas? Las tablas organizan la información para que pueda ser consultada con facilidad. Por esta razón, para crear las bases de datos debemos tener un plan.

¿Cuál será el contenido de la tabla? ¿Para qué queremos guardar la información? ¿Cuándo la consultaremos?, son algunas de las cuestiones previas al diseño a tener en cuenta.

1. Trabajar con un ejemplo delineado a partir de las preguntas arriba citadas:

Debemos guardar los datos de los clientes de una empresa y las llamadas que ellos hacen por diversos motivos, por ejemplo para pedir soporte técnico.

2. Definir cuestiones como:

Cómo distinguir a cada cliente: a través de un número o código. Ese código lo ubicará sin dudas aunque tengan nombres, apellidos o domicilios similares. Para ello usamos un número único llamado ID (Identificador) para cada cliente ajeno a los datos inherentes a la persona.

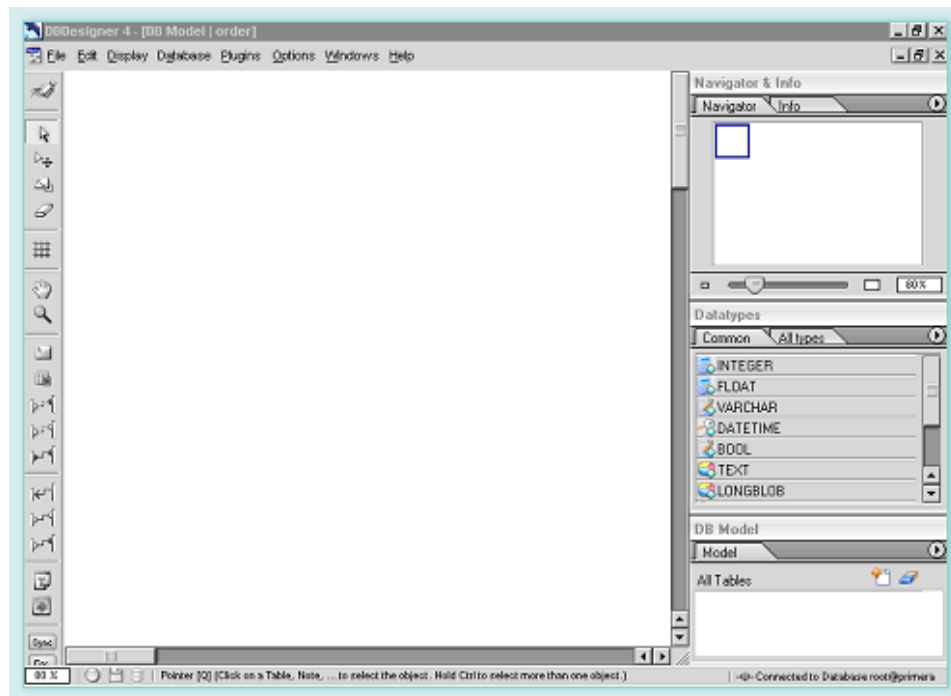
(Algunas veces, se utiliza como código un dato que pertenece a la misma información que se está archivando, como por ejemplo el DNI, pero esto trae problemas, en el caso por ejemplo de que sea mal ingresado.)

3. Definir los datos de la base:

- ID del cliente
- Documento
- Apellido
- Nombres

- Dirección
- Localidad
- Provincia
- Fecha de ingreso
- Saldo cuenta corriente

4. Comprobar la conexión con la base primera en el "localhost":

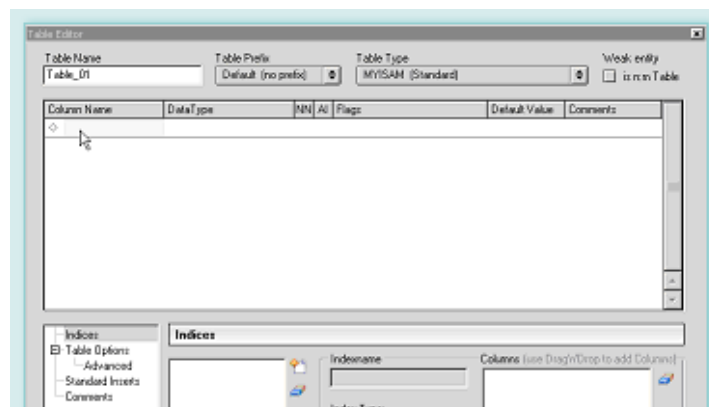


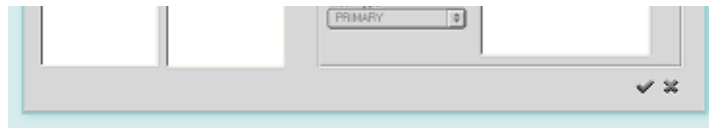
La última línea de estado indica que estamos como root en la base de datos "primera".

5. En la parte inferior del escritorio -donde dice "All tables"- seleccionar el ícono con el signo + en color naranja.

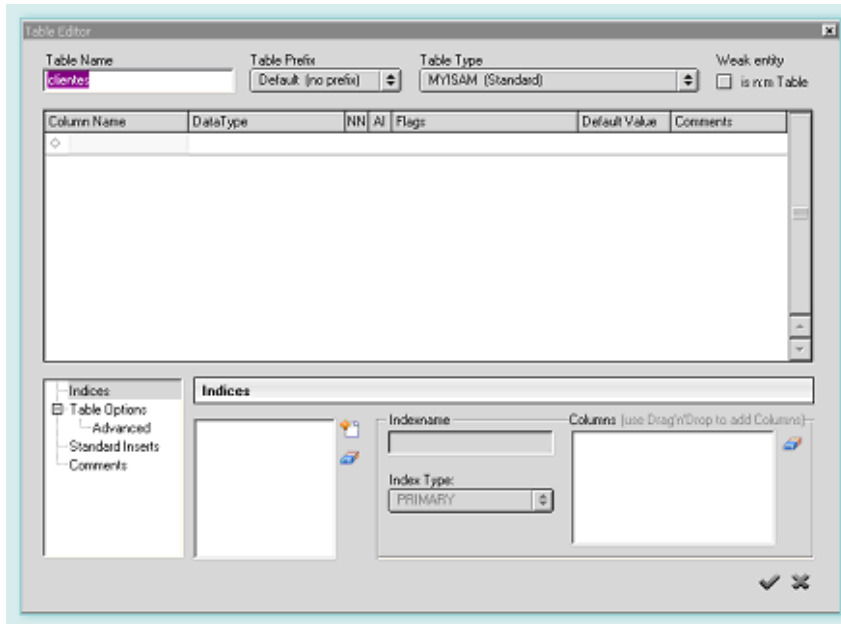


Aparecerá en la pantalla de diseño de la tabla con valores por defecto -como el nombre de la tabla: "Tabla_01"- y todo lo demás, en blanco.

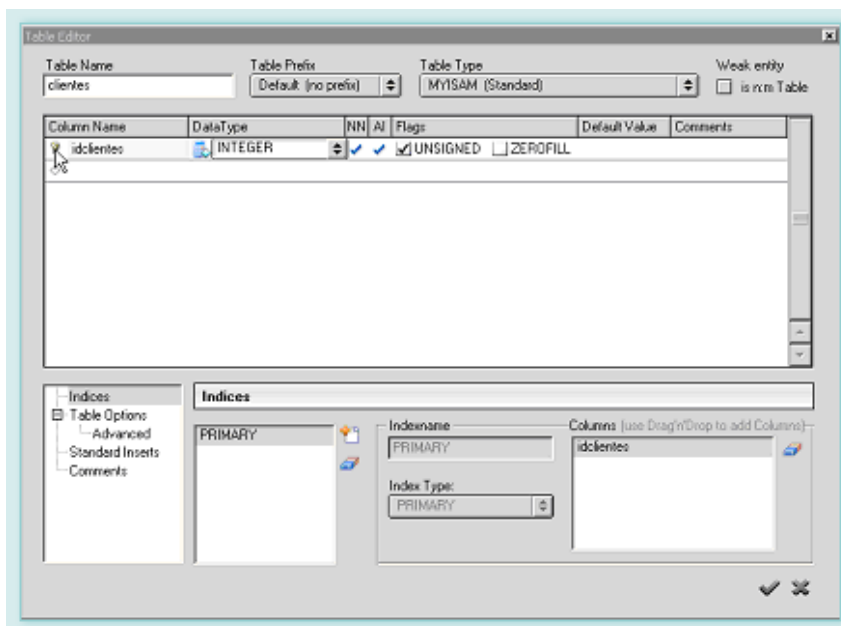




6. Cambiar el nombre de la tabla por otro, por ejemplo, "clientes".



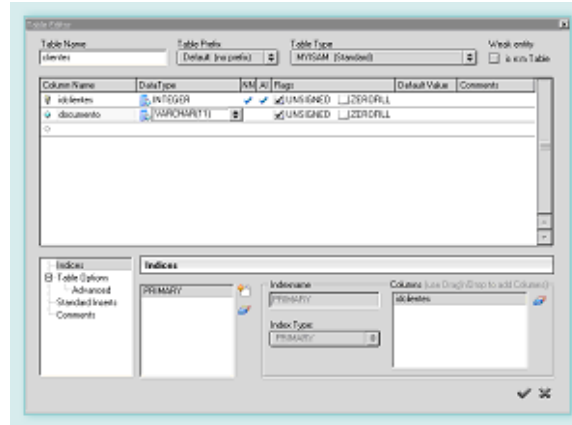
7. Seleccionar el primer renglón. Aparecerá automáticamente el campo "idcliente" con ciertas características:



- Tipo INTEGER
- No puede ser NIL (NN)
- Autoincremental (AI)
- Sin Signo (UNNSIGNED)

Este campo es un campo clave que MySQL administra automáticamente al agregarse un nuevo cliente. Se la denomina **clave**

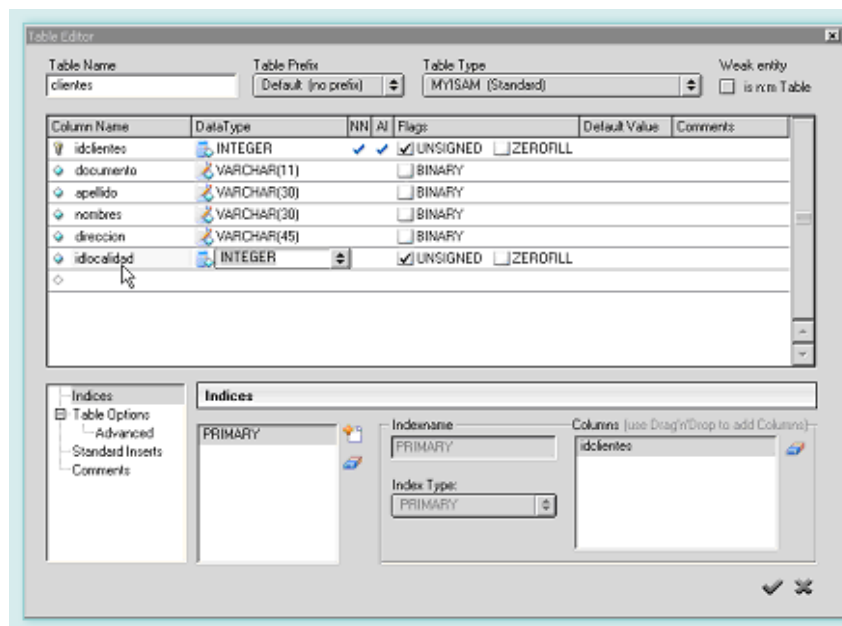
primaria, es única para cada cliente y se mantiene aunque cambien todos los demás datos.
El campo o columna de la base de datos queda entonces así:



8. Agregar el campo Documento a través de los siguientes pasos: - Seleccionar el tipo de campo VARCHAR de 11 posiciones. Escribir entre paréntesis el valor 11 "para indicarle al programa MySQL" que el campo tiene 11 caracteres.

En MySQL la longitud del campo indica los caracteres que serán mostrados y no su longitud máxima -que en el caso del VARCHAR es de 255 caracteres-. Definimos 11 caracteres porque el CUIT tiene dos números más el número de documento de ocho posiciones y el dígito verificador de un carácter.

¿Por qué no usar simplemente un entero (INTEGER) como campo? Porque este campo o columna no va a intervenir en cálculo numérico alguno y los campos de carácter son más sencillos de manipular.



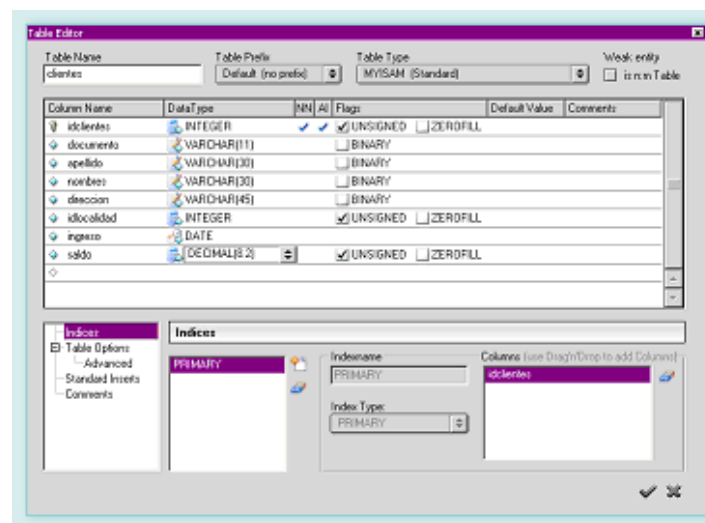
9. Con el mismo esquema, agregar:

El campo Apellido seleccionando el tipo VARCHAR e indicando una longitud de 30 caracteres

El campo Nombres, también VARCHAR, de longitud 30 caracteres.

El campo Dirección, de longitud 45 (no separamos la calle del número de la puerta)

10. Crear los campos Localidad y Provincia. En este caso, se puede aprovechar el poder de la base de datos relacional y usar el ID de una tabla de localidades que se debe crear luego. Esto evitaría tener que escribir cada vez el registro y disminuiría el riesgo de errores que dificultarían las búsquedas.

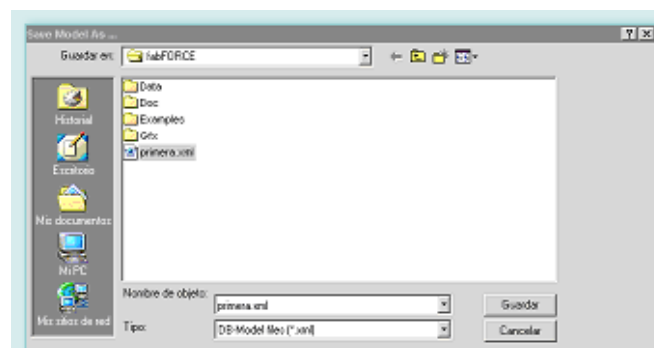


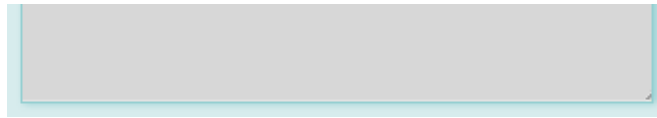
11. Crear el campo Fecha de ingreso. Se recomienda utilizar el tipo de fecha DATE que permite realizar cálculos con facilidad.

12. Crear el campo Saldo cuenta corriente utilizando el campo tipo FLOAT o DECIMAL -que son equivalentes- con una longitud de 8 para la mantisa y dos decimales.

Los nombres de los campos deben cumplir con ciertas reglas: deben empezar con letras, no admiten espacios en ellos y algunos otros detalles (véase por favor el manual de MySQL).

Una vez que se hace clic en la tilde color gris, significa que el trabajo está aceptado y se obtiene la siguiente pantalla:

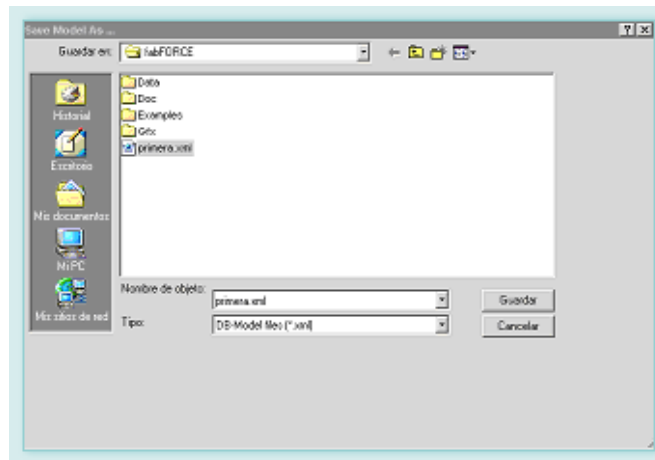




13. Salvar el modelo con el comando:

File->Save as...

Aparece entonces el cuadro de diálogo:

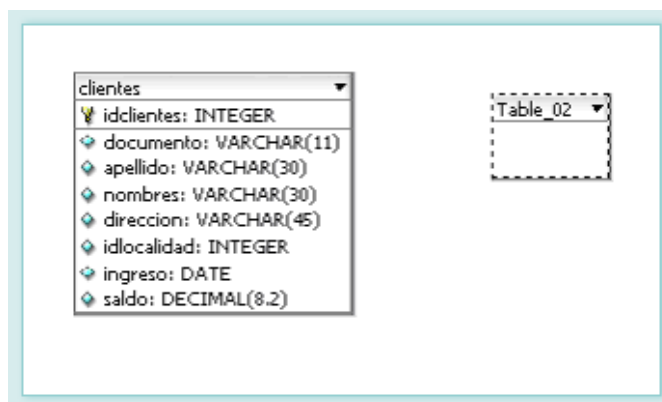


Anotar "primera" como nombre del modelo y aceptar.

Etapa 4. Diseñar la tabla de localidades

Para realizar esta tarea, vamos a usar un método distinto del que usamos para los clientes.

1. Al lado de la tabla clientes, pulsar las teclas Ctrl y T simultáneamente. Aparece entonces una "tabla vacía" como la de este dibujo:



2. Hacer clic dentro del cuadrado y reaparecerá la caja de diseño de la tabla que usamos para construir la de clientes.

3. Organizar una tabla de localidades simple, con tres campos:

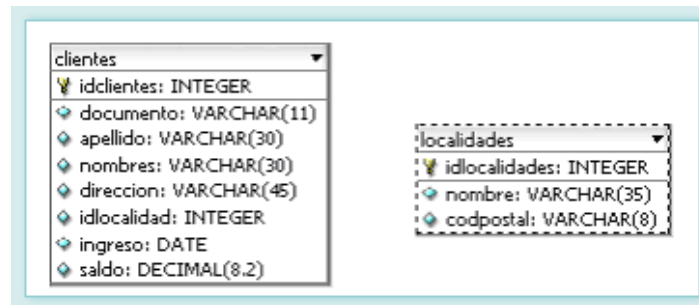
- Idlocalidad
- Nombre
- Código postal

idlocalidad es el campo clave con que relacionaremos esta tabla con la de clientes; como siempre es INTEGER AUTOINCREMENTAL NOT NULL.

nombre es un campo VARCHAR de 35 posiciones.

código postal es un campo VARCHAR de 8 posiciones.

La tabla queda entonces de la siguiente manera:



Etapa 5. Armar relaciones con la tabla provincias

1. Crear la tabla con las columnas:

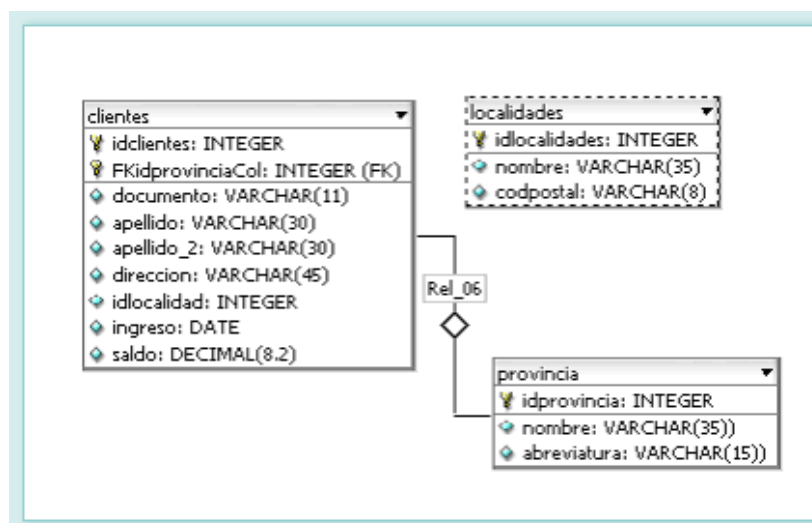
- Idprovincia Integer Not Null Autonumeric
- Nombre Varchar de 35
- Abreviatura Varchar de 15

2. Agregar a la tabla de clientes una relación 1:1 (uno a uno) de la tabla de clientes a la de provincias. Esto es porque a cada cliente le corresponde una sola provincia.

3. Seleccionar el ícono que está en la regla del costado:



4. Seleccionar la tabla de provincias y luego la de clientes y automáticamente DBDesigner creará una relación y un campo más a la tabla de clientes.



5. Salvar el modelo con la opción File->Save.

6. Pasar el modelo a MySQL. Para esto crear las tablas e índices en el motor de base de datos relacionales a través de los comandos File->Export->SQL Script...

Tal como se verá a continuación, hay dos opciones para este resultado:

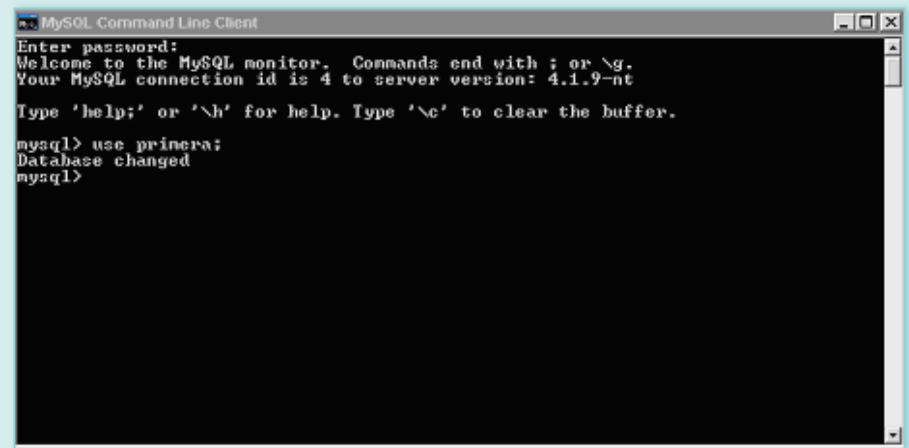
a. Pasar el script al portapapeles: Copy script to clipboard.

b. Grabarlo a un archivo Save script to file.

7. Seleccionar Copy script to clipboard.

8. Abrir el monitor de MySQL nuevamente: Inicio->Programas->MySQL->MySQL Server 4.1->MySQL Command Line Client.

Aparece nuevamente la pantalla siguiente. Escribir la palabra clave y luego en la petición seleccionar la base de datos primera con la orden: use primera; seguido de lo que vemos a continuación:

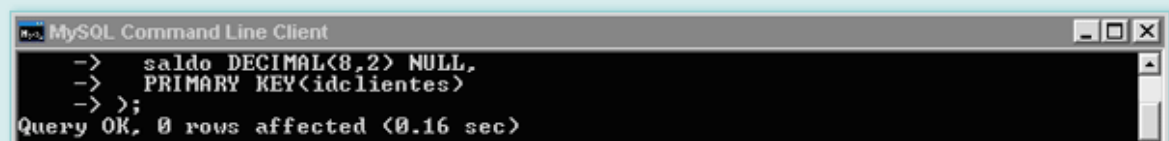


```
MySQL Command Line Client
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 4.1.9-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use primera;
Database changed
mysql>
```

9. "Pegar" el script SQL mediante el menú que aparece al hacer clic en el pequeño icono que está en el extremo superior derecho de la ventana. Seleccionar Editar->Pegar y aparece la siguiente pantalla .



```
MySQL Command Line Client
-> saldo DECIMAL(8,2) NULL,
-> PRIMARY KEY(idclientes)
-> >;
Query OK, 0 rows affected (0.16 sec)
```

```

mysql>
mysql> CREATE TABLE localidades (
-> idlocalidades INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
-> nombre VARCHAR(35) NULL,
-> codpostal VARCHAR(8) NULL,
-> PRIMARY KEY(idlocalidades)
-> );
Query OK, 0 rows affected (0.15 sec)

mysql>
mysql> CREATE TABLE provincia (
-> idprovincia INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
-> nombre VARCHAR(35) NULL,
-> abreviatura VARCHAR(15) NULL,
-> PRIMARY KEY(idprovincia)
-> );
Query OK, 0 rows affected (0.16 sec)

mysql>
mysql>

```

Se ejecutaron en el monitor de MySQL las siguientes líneas:

```

CREATE TABLE clientes (
idclientes INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
FKidprovinciaCol INTEGER UNSIGNED NOT NULL,
documento VARCHAR(11) NULL,
apellido VARCHAR(30) NULL,
apellido_2 VARCHAR(30) NULL,
direccion VARCHAR(45) NULL,
idlocalidad INTEGER UNSIGNED NULL,
ingreso DATE NULL,
saldo DECIMAL(8,2) NULL,
PRIMARY KEY(idclientes);
)

```

```

CREATE TABLE localidades (
idlocalidades INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
nombre VARCHAR(35) NULL,
codpostal VARCHAR(8) NULL,
PRIMARY KEY(idlocalidades)
)

```

```

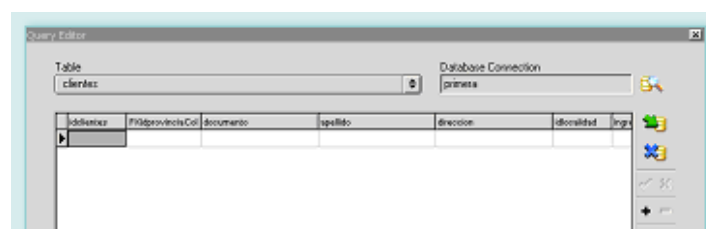
CREATE TABLE provincia (
idprovincia INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
nombre VARCHAR(35) NULL,
abreviatura VARCHAR(15) NULL,
PRIMARY KEY(idprovincia)
)

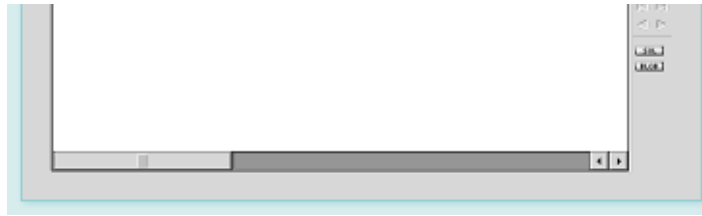
```

Con esto tenemos en nuestra base de datos MySQL las tablas creadas y listas para usarse.

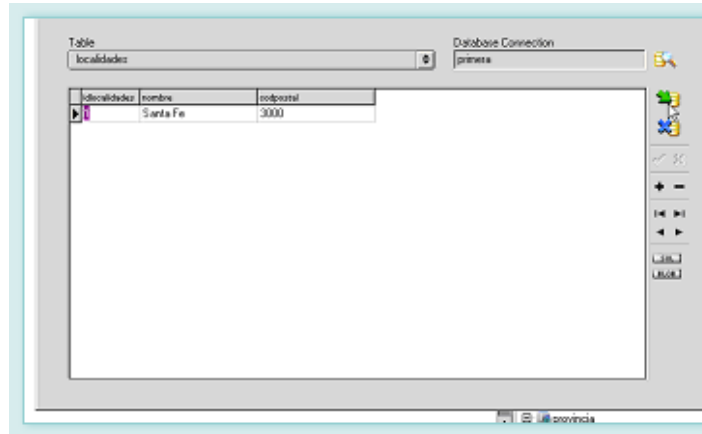
Etapas 6. Ingresar datos

1. Utilizar el DBDesigner haciendo clic con el botón derecho del mouse sobre la tabla clientes.
2. Seleccionar Edit Table Data del menú contextual y aparece:



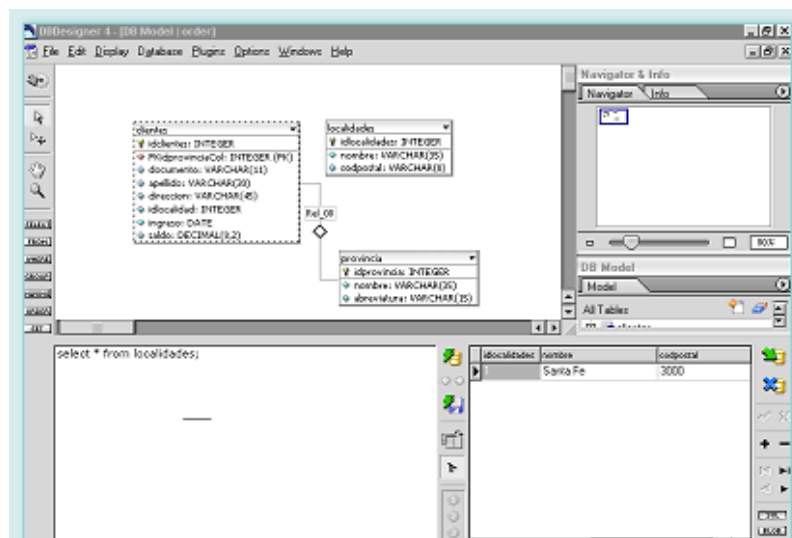


3. Seleccionar localidades desde la lista de selección de las tablas y luego agregar un registro:



4. Con esta poderosa herramienta es posible usar el SQL en forma directa. Para ello pasar del modo Diseño al modo SQL desde el menú: Display->Query Mode.

Por ejemplo:



[ACERCA DE ...](#)[CÓMO USAR EL CD](#)[MAPA DEL CD](#)[LIC. CREATIVE COMMONS](#)[CRÉDITOS](#)