1.- Campo Entero con Autoincremento.

Un campo de tipo entero puede tener otro atributo extra 'auto_increment'. Los valores de un campo 'auto_increment', se inician en 1 y se incrementan en 1 automáticamente.

Se utiliza generalmente en campos correspondientes a códigos de identificación para generar valores únicos para cada nuevo registro que se inserta.

Sólo puede haber un campo "auto_increment" y debe ser clave primaria (o estar indexado).

Para establecer que un campo autoincremente sus valores automáticamente, éste debe ser entero (integer) y debe ser clave primaria:

```
create table libros(
 codigo int auto_increment,
 titulo varchar(20),
 autor varchar(30),
 editorial varchar(15),
 primary key (codigo)
);
```

Para definir un campo autoincrementable colocamos "auto_increment" luego de la definición del campo al crear la tabla.

Hasta ahora, al ingresar registros, colocamos el nombre de todos los campos antes de los valores; es posible ingresar valores para algunos de los campos de la tabla, pero recuerde que al ingresar los valores debemos tener en cuenta los campos que detallamos y el orden en que lo hacemos.

Cuando un campo tiene el atributo "auto_increment" no es necesario ingresar valor para él, porque se inserta automáticamente tomando el último valor como referencia, o 1 si es el primero.

Para ingresar registros omitimos el campo definido como "auto_increment", por ejemplo:

```
insert into libros (titulo,autor,editorial)
 values('El aleph','Borges','Planeta');
```

Este primer registro ingresado guardará el valor 1 en el campo correspondiente al código.

Si continuamos ingresando registros, el código (dato que no ingresamos) se cargará automáticamente siguiendo la secuencia de autoincremento.

Un campo "auto_increment" funciona correctamente sólo cuando contiene únicamente valores positivos. Más adelante explicaremos cómo definir un campo con sólo valores positivos.

Está permitido ingresar el valor correspondiente al campo "auto_increment", por ejemplo:

```
 insert into libros (codigo,titulo,autor,editorial)
  values(6,'Martin Fierro','Jose Hernandez','Paidos');
```

Pero debemos tener cuidado con la inserción de un dato en campos "auto_increment". Debemos tener en cuenta que:

```
- si el valor está repetido aparecerá un mensaje de error y el
registro no se
  ingresará.
- si el valor dado saltea la secuencia, lo toma igualmente y
en las siguientes
  inserciones, continuará la secuencia tomando el valor más
alto.
- si el valor ingresado es 0, no lo toma y guarda el registro
continuando la
  secuencia.
- si el valor ingresado es negativo (y el campo no está
definido para aceptar sólo
  valores positivos), lo ingresa.
```

1.- Ejemplo

Una farmacia guarda información referente a sus medicamentos en una tabla llamada "medicamentos".

1- Elimine la tabla,si existe:

```
 drop table if exists medicamentos;
```

2- Cree la tabla con la siguiente estructura:

```
 create table medicamentos(
   codigo integer auto_increment,
   nombre varchar(20),
   laboratorio varchar(20),
   precio float,
   cantidad integer,
   primary key (codigo)
);
```

3- Visualice la estructura de la tabla "medicamentos" (describe).

4- Ingrese los siguientes registros (insert into):

```
 insert into medicamentos (nombre, laboratorio,precio,cantidad)
    values('Sertal','Roche',5.2,100);
 insert into medicamentos (nombre, laboratorio,precio,cantidad)
   values('Buscapina','Roche',4.10,200);
 insert into medicamentos (nombre, laboratorio,precio,cantidad)
   values('Amoxidal 500','Bayer',15.60,100);
```

5- Verifique que el campo "código" generó los valores de modo automático:

```
 select codigo,nombre,laboratorio,precio,cantidad
  from medicamentos;
```

6- Intente ingresar un registro con un valor de clave primaria repetido.

7- Ingrese un registro con un valor de clave primaria no repetido salteando la secuencia:

```
 insert into medicamentos (codigo,nombre, laboratorio,precio,cantidad)
  values(12,'Paracetamol 500','Bago',1.90,200);
```

8- Ingrese el siguiente registro:

```
 insert into medicamentos (nombre, laboratorio,precio,cantidad)
  values('Bayaspirina','Bayer',2.10,150);
```

2.- Valores por defecto.

Si al insertar registros no se especifica un valor para un campo, se inserta su valor por defecto implícito según el tipo de dato del campo. Por ejemplo:

```
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Java en 10 minutos','JuanPereyra','Paidos',25.7,100);
```

Como no ingresamos valor para el campo "codigo", MySQL insertará el valor por defecto, como "codigo" es un campo "auto_increment", el valor por defecto es el siguiente de la secuencia.

Si omitimos el valor correspondiente al autor:

```
insert into libros (titulo,editorial,precio,cantidad)
values('Java en 10 minutos','Paidos',25.7,200);
```

MySQL insertará "null", porque el valor por defecto de un campo (de cualquier tipo) que acepta valores nulos, es "null".

Lo mismo sucede si no ingresamos el valor del precio:

```
insert into libros (titulo,autor,editorial,cantidad)
 values('Java en 10 minutos','Juan Pereyra',Paidos',150);
```

MySQL insertará el valor "null" porque el valor por defecto de un campo (de cualquier tipo) que acepta valores nulos, es "null".

Si omitimos el valor correspondiente al título:

```
insert into libros (autor,editorial,precio,cantidad)
 values ('Borges','Paidos',25.7,200);
```

MySQL guardará una cadena vacía, ya que éste es el valor por defecto de un campo de tipo cadena definido como "not null" (no acepta valores nulos).

Si omitimos el valor correspondiente a la cantidad:

```
insert into libros (titulo,autor,editorial,precio)
 values('Alicia a traves del espejo','Lewis
Carroll','Emece',34.5);
```

el valor que se almacenará será 0, porque el campo "precio" es de tipo numérico "not null" y el valor por defecto de los tipos numéricos que no aceptan valores nulos es 0.

Podemos establecer valores por defecto para los campos cuando creamos la tabla. Para ello utilizamos "default" al definir el campo. Por ejemplo, queremos que el valor por defecto del campo "precio" sea 1.11 y el valor por defecto del campo "autor" sea "Desconocido":

```
create table libros(
 codigo int unsigned auto_increment,
 titulo varchar(40) not null,
 autor varchar(30) default 'Desconocido',
 precio decimal(5,2) unsigned default 1.11,
 cantidad int unsigned not null,
 primary key (codigo)
);
```

Si al ingresar un nuevo registro omitimos los valores para el campo "autor" y "precio", MySQL insertará los valores por defecto definidos con la palabra clave "default":

```
insert into libros (titulo,editorial,cantidad)
 values('Java en 10 minutos','Paidos',200);
```

MySQL insertará el registro con el siguiente valor de la secuencia en "codigo", con el título, editorial y cantidad ingresados, en "autor" colocará "Desconocido" y en precio "1.11".

Entonces, si al definir el campo explicitamos un valor mediante la cláusula "default", ése será el valor por defecto; sino insertará el valor por defecto implícito según el tipo de dato del campo.

Los campos definidos "auto_increment" no pueden explicitar un valor con "default", tampoco los de tipo "blob" y "text".

Los valores por defecto implícitos son los siguientes:

-para campos de cualquier tipo que admiten valores nulos, el valor por defecto "null";

-para campos que no admiten valores nulos, es decir, definidos "not null", el valor por defecto depende del tipo de dato:

-para campos numéricos no declarados "auto_increment": 0;

-para campos numéricos definidos "auto_increment": el valor siguiente de la secuencia, comenzando en 1;

-para los tipos cadena: cadena vacía.

Ahora al visualizar la estructura de la tabla con "describe" podemos entender un poco más lo que informa cada columna:

```
describe libros;
```

"Field" contiene el nombre del campo; "Type", el tipo de dato; "NULL" indica si el campo admite valores nulos; "Key" indica si el campo está indexado (lo veremos más adelante); "Default" muestra el valor por defecto del campo y "Extra" muestra información adicional respecto al campo, por ejemplo, aquí indica que "codigo" está definido "auto_increment".

También se puede utilizar "default" para dar el valor por defecto a los campos en sentencias "insert", por ejemplo:

```
insert into libros (titulo,autor,precio,cantidad)
 values ('El gato con botas',default,default,100);
```

EJERCICICIO

Un comercio que envía pizzas y empanadas a domicilio registra los pedidos diariamente en una tabla llamada "pedidos" con los siguientes datos:
- numero de pedido, autoincrementable, entero positivo comienza en 1 y menor a 200 aprox.
- nombre: pizza o empanada, por defecto "empanada",
- tipo: por ejemplo, si es pizza: especial, muzarela, etc., si son empanadas: arabes, pollo, jamón
y queso, criollas, etc.
- precio: precio por unidad, valor con decimales que no supera los $99.99 y será siempre mayor a
0, por defecto "1",
- cantidad: cantidad de articulos, entero positivo desde 1 e inferior a 200 aprox., por defecto "12"
- domicilio del cliente.

1- Elimine la tabla "pedidos" si existe.

2- Cree la tabla eligiendo el tipo de dato adecuado para cada campo.

3- Ingrese los siguientes registros:

```
insert into pedidos (nombre,tipo,precio,cantidad,domicilio)
  values('piza','muzarela','4.00',3,'Sarmiento 235');
insert into pedidos (tipo,precio,cantidad,domicilio)
  values('arabe','1.00',24,'Urquiza 296');
insert into pedidos (nombre,tipo,domicilio)
  values('empanada','salteña','Colon 309');
insert into pedidos (tipo,domicilio)
  values('arabe','San Martin 444');
insert into pedidos (nombre,tipo,precio,domicilio)
  values('piza','especial','4.00','Avellaneda 395');
```

4- Muestre todos los campos de todos los pedidos para ver cómo se guardaron los datos no ingresados.