
Migraciones o cambios de esquema

Cuando un programa utiliza una base de datos, este trabaja directamente sobre los datos de la base: insertando, actualizando, buscando y borrando objetos en las distintas tablas.

Entonces las tablas que utiliza el programa deben crearse antes de que el programa empiece a operar con la base de datos. Las mismas se pueden crear con un *script sql* que se ejecute en la base de datos para prepararla y que el programa pueda comenzar a operar sobre ella.

Las migraciones o cambios de esquema se guardan para poder tener un histórico de la base de datos y poder pasar de una versión del programa que utilizaba cierta versión de la base de datos, a otra versión del programa que utiliza otra versión de la base de datos.

Entonces una primera migración de una base de datos para un programa que trabaja con monedas podría ser esta:

```
BEGIN TRANSACTION;

CREATE TABLE IF NOT EXISTS 'currency' (
    'code'    TEXT,
    'name'    text,
    'symbol' text,
    PRIMARY KEY('code')
);

COMMIT;
```

Supongamos que tenemos en producción nuestro programa que implementa la clase *Currency*, con las variables de instancia *code*, *name* y *symbol*. Al utilizarlo se van agregando objetos de tipo *Currency* en la base de datos y operando con ellos.

Si en un futuro se agrega una nueva variable de instancia al objeto *Currency*, por ejemplo, la variable *factor*. No puedo instalar el cambio de código sin antes hacer una modificación a la base de datos. Y si el programa tiene muchas implementaciones, es conveniente generar cambios de esquemas, como para poder pasar de una versión del programa a otra sin problemas.

Entonces en el ejemplo mencionado, tendremos que crear una migración que agregue el campo *factor* a la tabla *currency* en la base de datos.

```
BEGIN TRANSACTION;  
  
ALTER TABLE 'currency'  
ADD 'factor' REAL;  
  
COMMIT;
```

En este caso el campo puede estar nulo, pero si lo definimos para que no sea nulo, tendríamos que ponerle algún valor por defecto al campo *factor* para cada registro. Eso también se hace en la migración, llamada migración de datos. Por ejemplo:

```
BEGIN TRANSACTION;  
  
ALTER TABLE 'currency'  
ADD 'factor' REAL NOT NULL;  
  
UPDATE currency SET factor = 1;  
  
COMMIT;
```

De esta manera, todos los registros que existan en la base de datos tendrán como valor de la columna factor el 1.

Do Not Copy or Post