PHP y SQLite

Tipos de columnas

Francisco Arce
www.pacoarce.com

SQLite Clases de almacenamiento:

Cada columna en SQLite tiene una de estas clases:

Clase	Descripción
NULL	NULL es considerado un valor vacío.
INTEGER	El valor entero con signo entre 1 y 8 bytes dependiendo de la magnitud del valor.
REAL	El valor es de punto flotante, se almacena como un número de coma flotante de 8 bytes IEEE.
TEXT	El valor es una cadena de texto, utilizando la codificación de la base de datos (UTF- 8, UTF - 16BE o UTF - 16LE)
BLOB	Permite almacenar objetos en formato BLOB (Binary Large Objects) como imágenes, archivos de sonido y otros objetos multimedia; a veces se almacenan como BLOB código de binarios.

PHP y SQLite

Affinity

Francisco Arce
www.pacoarce.com

SQLite tipo de afinidad o Affinity Type:

En SQLite contiene un concepto llamado “afinidad” o “affinity”. Cada columna puede seguir almacenando en su tipo de columna, pero puede ser representado con diferente “máscara”, con lo cual lo hace “a fin” con otras bases de datos.

Cada columna en SQLite 3 tiene una clase y una afinidad:

Affinity: INTEGER

Data Type

- INT
- INTEGER
- TINYINT
- SMALLINT
- MEDIUMINT
- BIGINT
- UNSIGNED BIG INT
- INT2
- INT8

Affinity: TEXT

Data Type

- CHARACTER(20)
- VARCHAR(255)
- VARYING CHARACTER(255)
- NCHAR(55)
- NATIVE CHARACTER(70)
- NVARCHAR(100)
- TEXT
- CLOB

Affinity: NONE

Data Type
BLOB
(No hay tipos de datos afines)

Affinity: REAL

Data Type

- REAL
- DOUBLE
- DOUBLE PRECISION
- FLOAT

Boolean Datatype:

SQLite Los valores boolenos no tienen subclases. 1 es verdadero y 0 falso.

Tipo de datos para fechas y horas (Date/Time):

SQLite no tiene subclases o afinidades para las fechas, pero se pueden almacenar en las clases TEXT, REAL o INTEGER.

Clases	Formato de fecha
TEXT	Formato "YYYY-MM-DD HH:MM:SS.SSS".
REAL	El número de días desde el mediodía en Greenwich el 24 de noviembre de 4714 aC
INTEGER	Número de segundos desde 1970-01-01 00:00:00 UTC a la fecha.

Puedes utilizar cualquier operación entre fechas sin importar las clases.

Francisco Arce
www.pacoarce.com

PHP y SQLite

Crear una tabla

Francisco Arce
www.pacoarce.com

```
CREATE TABLE database_name.table_name(
   column1 datatype,
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype,
);
```

Versión 3.3+

```
create table if not exists tabla(col1 typ1, ..., colN typN)
```

Francisco Arce
www.pacoarce.com

PHP y SQLite

Eliminar una tabla

Francisco Arce
www.pacoarce.com

En SQLite, con la sentencia DROP TABLE se elimina una tabla con su definición, datos asociados, índices, triggers, constrains y permisos.

Cuando una tabla es eliminada, no hay forma de recuperarla. Se borra en forma definitiva.

Sintaxis:

DROP TABLE basededatos.tabla;

En versión 3.3. y más:

drop table if exists TableName

PHP y SQLite

Constraints

Francisco Arce
www.pacoarce.com

Modifica la columna de la tabla. Ejemplos de comandos “constraints”:

NOT NULL
PRIMARY KEY
UNIQUE
AUTOINCREMENT
DEFAULT
CURRENT_TIME
CURRENT_DATE
CURRENT_TIMESTAMP

PHP y SQLite

Constraints a nivel tablas

Francisco Arce
www.pacoarce.com

Constraints a nivel tabla

Podemos escribir “*constraints*” a nivel tabla.

Por lo general involucra a más de una columna, pero puede involucrar sólo una sin problema.

Generalmente utilizamos los constraint PRIMARY KEY, UNIQUE y CHECK a nivel tabla cuando involucran a más de una columna.

PHP y SQLite

Crear una tabla desde un query

Francisco Arce
www.pacoarce.com

Crear una tabla desde un query

Podemos crear tabla desde un query o la sentencia select.

CREATE [TEMP] TABLE nombreTabla AS SELECT query;

El *query* sólo se ejecuta una vez, cuando se crea la tabla.

Las tablas temporales solo existen cuando la base de datos esté abierta. Cuando se cierra, desaparecen.

Las tablas temporales sólo pueden ser accesadas por la conexión con la que fueron creadas.

Francisco Arce
www.pacoarce.com

PHP y SQLite

Alterar una tabla

Francisco Arce
www.pacoarce.com

Alterar o modificar una tabla

En SQLite la sentencia ALTER TABLE solo puede añadir columnas y renombrar la tabla.

No podemos eliminar columnas con esta sentencia.

Las nuevas columnas siempre se añaden al final de la tabla.

Si necesitas hacer muchas modificaciones, es mejor hacer una tabla nueva y copiar los datos con un SELECT.

PHP y SQLite

Primary key

Francisco Arce
www.pacoarce.com

Primary key

- Siempre es importante declarar una o varias columnas como Primary Key, que es el índice principal de la tabla.
- Sólo podemos tener un índice principal por tabla.
- Las llaves en un índice principal deben ser únicas, es decir, no repetirse.
- El *constraint* “UNIQUE” queda implicito en “PRIMARY KEY”.

Primary key

- El constraint “NOT NULL” no está implícito en “PRIMARY KEY”. Al menos una columna debe contener el “NOT NULL”.
- Si generamos un índice principal como “INTEGER PRIMARY KEY” queda como equivalente al “ROWID” de la tabla.
- Podemos añadir “AUTOINCREMENT” a un índice principal entero.
- Sólo podemos tener una columna marcada como “AUTOINCREMENT”.

PHP y SQLite

Las vistas

Francisco Arce
www.pacoarce.com

Las vistas o views

- Las vistas o views proporcionan una manera de empaquetar consultas en un objeto predefinido.
- Una vez creadas, las vistas actúan de como tablas de sólo lectura.
- Al igual que las tablas, las vistas pueden ser temporales.
- La sintaxis básica del comando CREATE VIEW es:

CREATE [TEMP] VIEW *nombreVista* AS SELECT *query*

Francisco Arce
www.pacoarce.com

Las vistas o views

- Las vistas son dinámicas. Si modificas los registros en la tabla original, los cambios son reflejados en la vista cuando es llamada.
- Las vistas se pueden considerar como “querys empacados”.
- Por lo general las vistas nos servirán para hacer tablas reducidas para controlar los campos para ciertas aplicaciones o usuarios.
- Para eliminar una vista utilizamos:

```
DROP VIEW nombreVista
```

- Borrar una vista no afecta en nada a las tablas origen

PHP y SQLite

Crear índices

Francisco Arce
www.pacoarce.com

Crear índices

Los índices son una manera rápida de encontrar información en la tabla.

La sintaxis para crear un índice, una vez que la tabla ha sido creada es:

```
CREATE [UNIQUE] INDEX nombre_index ON nombre_tabla ( columna1 [, ...] );
```

Ni UNIQUE ni PRIMARY KEY implica que no se acepten valores nulos.

Crear índices

Para borrar un índice utilizamos:

```
DROP INDEX nombre_index;
```

No altera en nada los valores de la tabla.

Una forma de evitar que el usuario introduzca valores duplicados, es crear un índice UNIQUE. Si lo eliminas ya no tendrás esta validación.

PHP y SQLite

TRUNCATE TABLE

Francisco Arce
www.pacoarce.com

TRUNCATE TABLE

SQLite no cuenta con la sentencia TRUNCATE TABLE , pero podemos simularlo con DELETE. También puede borrar la tabla con DROP TABLE y volverla a crear..

Ejemplos:

DELETE FROM nombre_tabla;

DROP TABLE nombre_tabla;
CREATE TABLE nombre_tabla(...);

Francisco Arce
www.pacoarce.com