

CHARACTER SET y COLLATION En MySQL

url: <http://www.stan.com.mx/topics/view/12>

date: 4 de junio de 2008 a las 1:40

Contenido

UTF8 vs Latin1	1
UTF8 A Latin1 Y Viceversa	2
utf8_spanish_ci vs utf8_general_ci	2
utf8_spanish_ci vs utf8_spanish2_ci	2
utf8_general_ci vs utf8_bin	4
SHOW CHARACTER SET y SHOW COLLATION	4
Advertencia	5

Si tienes dudas sobre qué es un conjunto de caracteres (CHARACTER SET) o una colación (COLLATION) debes empezar leyendo Character Sets and Collations in General.

UTF8 vs Latin1

El latin1 es una codificación de 8 bits de longitud, el utf8 es una codificación de 8 o 16 o 24 bits de longitud. Obviamente el utf8 soporta una mayor cantidad de caracteres internacionales.



Ejemplo con una tabla latin 1:

```
CREATE TABLE ejemplolatin1 (  
  x varchar(10)  
) CHARACTER SET='latin1';  
  
INSERT INTO ejemplolatin1 VALUES ('pais');  
INSERT INTO ejemplolatin1 VALUES ('país');
```

Como resultado tenemos palabras de 4 bytes:

```
SELECT x, LENGTH(x), CHAR_LENGTH(x), HEX(x) FROM ejemplolatin1;  
  
-- 'pais', 4, 4, '70616973'  
-- 'país', 4, 4, '7061ED73'
```

Lo mismo pero con una tabla utf8:

```
CREATE TABLE ejemploutf8 (  
  x varchar(10)  
) CHARACTER SET='utf8';  
  
INSERT INTO ejemploutf8 VALUES ('pais');  
INSERT INTO ejemploutf8 VALUES ('país');
```

Ahora tenemos como resultado palabras de 4 y 5 bytes:

```
SELECT x, LENGTH(x), CHAR_LENGTH(x), HEX(x) FROM ejemploutf8;

-- 'pais', 4, 4, '70616973'
-- 'país', 5, 4, '7061C3AD73'
```

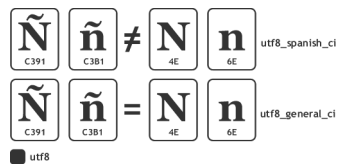
La función HEX() nos devuelve la cadena en hexadecimal. Podemos ver en la palabra país los valores en hexadecimal de í en latin1 y utf8 respectivamente.

UTF8 A Latin1 Y Viceversa

Para convertir entre codificaciones, tenemos que escribir _utf8 o _latin1 a cualquier cadena o expresión que represente una cadena:

```
SELECT _utf8'país', _latin1'país';
```

utf8_spanish_ci vs utf8_general_ci



Las siguientes comparaciones dan verdadero:

```
SELECT 'a' = 'A' COLLATE utf8_spanish_ci; -- 1
SELECT 'a' = 'á' COLLATE utf8_spanish_ci; -- 1
SELECT 'a' = 'à' COLLATE utf8_spanish_ci; -- 1
SELECT 'a' = 'ä' COLLATE utf8_spanish_ci; -- 1
SELECT 'a' = 'â' COLLATE utf8_spanish_ci; -- 1
```

Y también estas:

```
SELECT 'a' = 'A' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'á' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'à' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'ä' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'â' COLLATE utf8_general_ci; -- 1
```

Pero la siguiente comparación da falso:

```
SELECT 'n' = 'ñ' COLLATE utf8_spanish_ci; -- 0
```

En español moderno n no es igual a ñ. Cosa que si sucede en utf8_general_ci:

```
SELECT 'n' = 'ñ' COLLATE utf8_general_ci; -- 1
```

utf8_spanish_ci vs utf8_spanish2_ci

Antes del año 1994, ch y ll se consideraban letras independientes, entonces utf8_spanish2_ci (español tradicional) ordena ch entre c y d y ll entre l y m, cosa que no sucede con utf8_spanish_ci (español moderno):

```

CREATE TABLE ejemploSpanish (
  x varchar(10)
) CHARACTER SET='utf8';

INSERT INTO ejemploSpanish VALUES ('culebra');
INSERT INTO ejemploSpanish VALUES ('chuleta');
INSERT INTO ejemploSpanish VALUES ('luchador');
INSERT INTO ejemploSpanish VALUES ('llanta');

```




```

SELECT * FROM ejemploSpanish ORDER BY x COLLATE utf8_spanish_ci;

-- 'chuleta'
-- 'culebra'
-- 'llanta'
-- 'luchador'

```




```

SELECT * FROM ejemploSpanish ORDER BY x COLLATE utf8_spanish2_ci;

-- 'culebra'
-- 'chuleta'
-- 'luchador'
-- 'llanta'

```

Algo similar sucede con el idioma checo (utf8_czech_ci) que considera ch una letra entre h e i.



```

CREATE TABLE ejemploCzech (
  x varchar(10)
) CHARACTER SET='utf8';

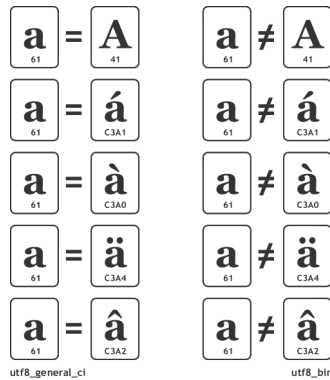
INSERT INTO ejemploCzech VALUES ('c');
INSERT INTO ejemploCzech VALUES ('h');
INSERT INTO ejemploCzech VALUES ('ch');
INSERT INTO ejemploCzech VALUES ('i');

SELECT * FROM ejemploCzech ORDER BY x COLLATE utf8_czech_ci;

-- 'c'
-- 'h'
-- 'ch'
-- 'i'

```

utf8_general_ci vs utf8_bin



Como vimos en los ejemplos de arriba, la colación utf8_spanish_ci no distingue entre:

- vocales con tilde (acento ortográfico) o sin tilde
- letras con diéresis o sin diéresis
- minúsculas y mayúsculas

La colación utf8_bin sí lo hace. Esta colación realiza una comparación binaria, bit por bit:

```
SELECT 'a' = 'A' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'á' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'à' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'ä' COLLATE utf8_general_ci; -- 1
SELECT 'a' = 'â' COLLATE utf8_general_ci; -- 1

SELECT 'a' = 'A' COLLATE utf8_bin; -- 0
SELECT 'a' = 'á' COLLATE utf8_bin; -- 0
SELECT 'a' = 'à' COLLATE utf8_bin; -- 0
SELECT 'a' = 'ä' COLLATE utf8_bin; -- 0
SELECT 'a' = 'â' COLLATE utf8_bin; -- 0
```

SHOW CHARACTER SET y SHOW COLLATION

La sentencia SHOW CHARACTER SET despliega todos los conjuntos de caracteres o codificaciones disponibles en el manejador MySQL:

```
SHOW CHARACTER SET;
```

La sentencia SHOW COLLATION despliega las diferentes colaciones disponibles en el manejador MySQL:

```
SHOW COLLATION;
```

Para conocer las codificaciones usadas actualmente por MySQL, la siguiente consulta puede ayudar:

```
SHOW VARIABLES LIKE '%char%';

-- 'character_set_client', 'utf8'
-- 'character_set_connection', 'utf8'
```

```
-- 'character_set_database', 'latin1'  
-- 'character_set_results', 'utf8'  
-- 'character_set_server', 'latin1'  
-- 'character_set_system', 'utf8'  
-- 'character_sets_dir', 'C:\mysql\share\charsets\'
```

Advertencia

La información de esta página no es confiable. El conocimiento se adquirió de forma empírica (o por fuerza bruta) y algunos términos pudieron ser inventados. Los trucos mencionados en este blog difícilmente son la manera más eficiente de resolver algún problema. La información no se actualiza y tampoco proviene de fuentes oficiales. Mejor acérquese a la documentación oficial, compre libros o visite la Wikipedia.