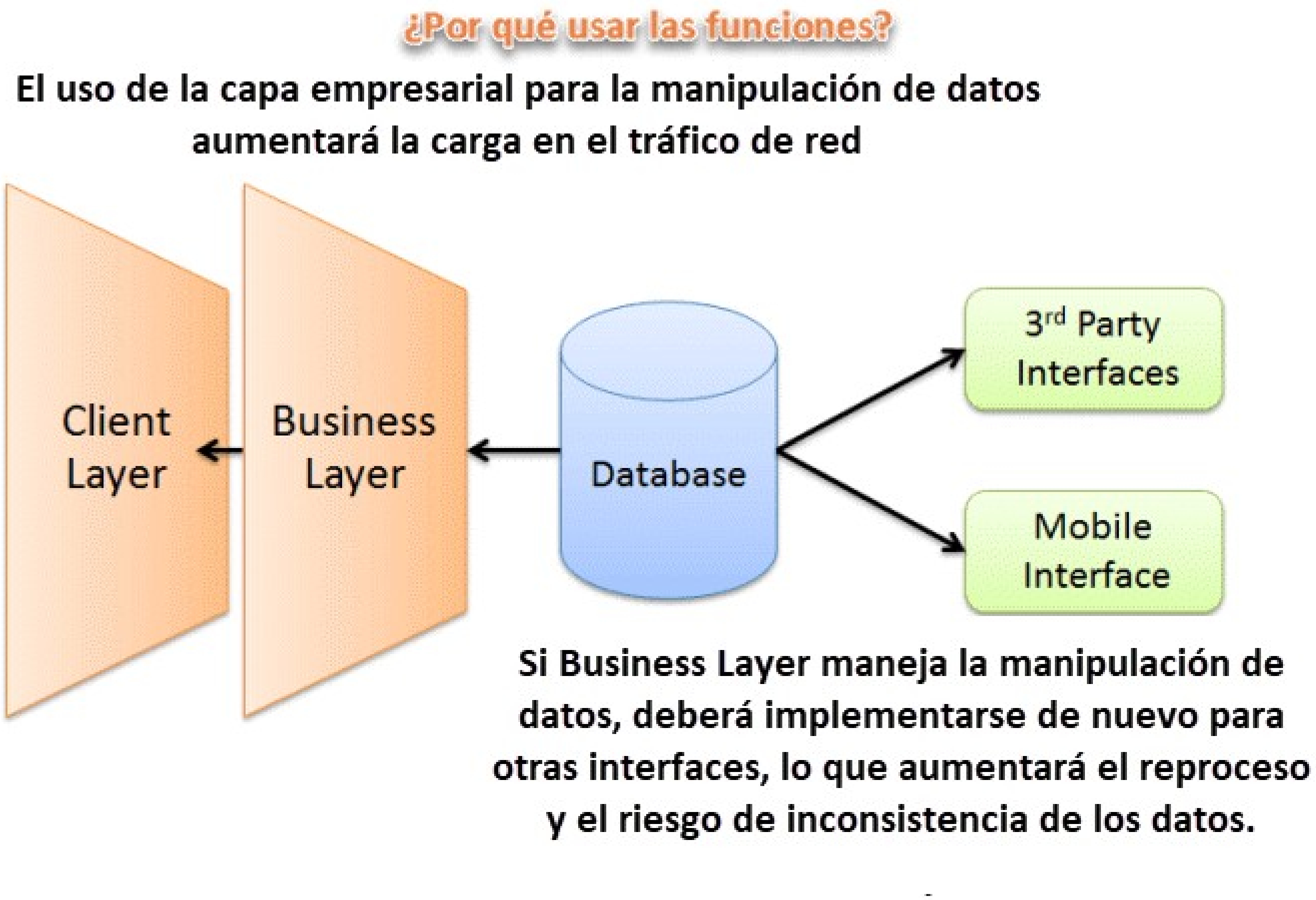
A3. Funciones propias del SGBD.

Índice.

A3. Funciones propias del SGBD.

1. ¿Por qué usar funciones?



¿Por qué usar las funciones?

El uso de la capa empresarial para la manipulación de datos aumentará la carga en el tráfico de red

Si Business Layer maneja la manipulación de datos, deberá implementarse de nuevo para otras interfaces, lo que aumentará el reproceso y el riesgo de inconsistencia de los datos.

A3. Funciones propias del SGBD.

1. ¿Por qué usar funciones?

MySQL puede hacer mucho más que almacenar y recuperar datos, sino también manipularlos antes de recuperarlos y guardarlos. Ahí es donde las funciones de MySQL tienen un gran juego.

Algunas funciones realizan operaciones con los datos y devuelven un resultado, otras sólo aceptar parámetros de entrada y otras no aceptan ningún tipo de parámetro.

La necesidad del uso de las funciones integradas en MySQL obedece a estas razones:

- Reducción del trabajo de la lógica comercial de la aplicación.
- Reducción de las inconsistencias de los datos.
- Ayuda en la reducción del tráfico de red en las aplicaciones cliente/servidor.

A3. Funciones propias del SGBD.

2. Clasificación de las funciones.

MySQL incorpora un amplio abanico de funciones que ya están implementadas en el servidor MySQL y que permiten realizar diferentes tipos de manipulaciones de datos.

Estas funciones integradas básicamente se pueden clasificar en las siguientes categorías:

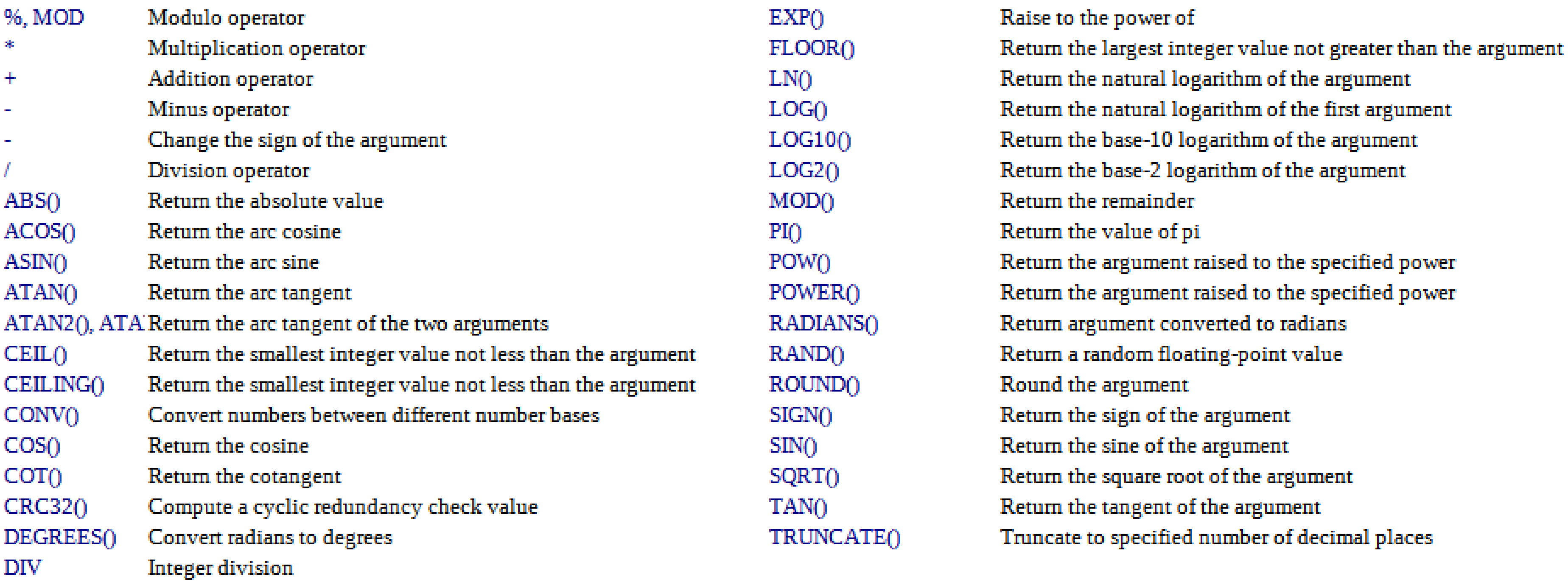
- Funciones de cadena → operan con tipos de datos de cadena.
- Funciones matermáticas → operan con tipos de datos numéricos.
- Funciones de fecha → operan con tipos de datos de fecha.
- Funciones agregadas → operan con los tipos de datos anteriores y ofrecen resultados resumidos.
- Funciones de control de flujo → permiten tomar decisiones y realizar bucles para obtener resultados.
- Otras funciones → permiten realizar tareas no incluidas anteriormente.

A3. Funciones propias del SGBD.

2.1. Funciones de cadena.

ASCII	Return numeric value of left-most character
BIN()	Return a string containing binary representation of a number
BIT_LENGTH(	Return length of argument in bits
CHAR()	Return the character for each integer passed
CHAR_LENGT	Return number of characters in argument
CHARACTER_	Synonym for CHAR_LENGTH()
CONCAT()	Return concatenated string
CONCAT_WS(	Return concatenate with separator
ELT()	Return string at index number
EXPORT_SET(	Return a string such that for every bit set in the value bits, you get an on string and for every unset bit, you get an off string
FIELD()	Index (position) of first argument in subsequent arguments
FIND_IN_SET(	Index (position) of first argument within second argument
FORMAT()	Return a number formatted to specified number of decimal places
FROM_BASE6	Decode base64 encoded string and return result
HEX()	Hexadecimal representation of decimal or string value
INSERT()	Insert substring at specified position up to specified number of characters
INSTR()	Return the index of the first occurrence of substring
LCASE()	Synonym for LOWER()
LEFT()	Return the leftmost number of characters as specified
LENGTH()	Return the length of a string in bytes
LIKE	Simple pattern matching
LOAD_FILE()	Load the named file
LOCATE()	Return the position of the first occurrence of substring
LOWER()	Return the argument in lowercase
LPAD()	Return the string argument, left-padded with the specified string
LTRIM()	Remove leading spaces
MAKE_SET()	Return a set of comma-separated strings that have the corresponding bit in bits set
MATCH()	Perform full-text search
MID()	Return a substring starting from the specified position
NOT LIKE	Negation of simple pattern matching
NOT REGEXP	Negation of REGEXP
OCT()	Return a string containing octal representation of a number
OCTET_LENGTH()	Synonym for LENGTH()
ORD()	Return character code for leftmost character of the argument
POSITION()	Synonym for LOCATE()
QUOTE()	Escape the argument for use in an SQL statement
REGEXP	Whether string matches regular expression
REGEXP_INSTR()	Starting index of substring matching regular expression
REGEXP_LIKE()	Whether string matches regular expression
REGEXP_REPLACE()	Replace substrings matching regular expression
REGEXP_SUBSTR()	Return substring matching regular expression
REPEAT()	Repeat a string the specified number of times
REPLACE()	Replace occurrences of a specified string
REVERSE()	Reverse the characters in a string
RIGHT()	Return the specified rightmost number of characters
RLIKE	Whether string matches regular expression
RPAD()	Append string the specified number of times
RTRIM()	Remove trailing spaces
SOUNDEX()	Return a soundex string
SOUNDS LIKE	Compare sounds
SPACE()	Return a string of the specified number of spaces
STRCMP()	Compare two strings
SUBSTR()	Return the substring as specified
SUBSTRING()	Return the substring as specified
SUBSTRING_INDEX()	Return a substring from a string before the specified number of occurrences of the delimiter
TO_BASE64()	Return the argument converted to a base-64 string
TRIM()	Remove leading and trailing spaces
UCASE()	Synonym for UPPER()
UNHEX()	Return a string containing hex representation of a number
UPPER()	Convert to uppercase
WEIGHT_STRING()	Return the weight string for a string

2.2. Funciones matemáticas.

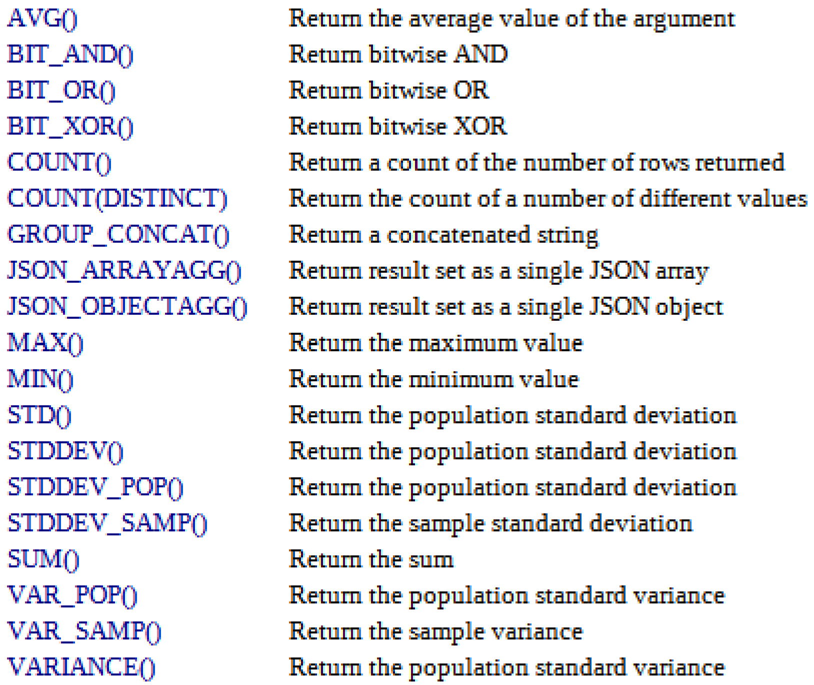


%, MOD	Modulo operator	EXP()	Raise to the power of
*	Multiplication operator	FLOOR()	Return the largest integer value not greater than the argument
+	Addition operator	LN()	Return the natural logarithm of the argument
-	Minus operator	LOG()	Return the natural logarithm of the first argument
-	Change the sign of the argument	LOG10()	Return the base-10 logarithm of the argument
/	Division operator	LOG2()	Return the base-2 logarithm of the argument
ABS()	Return the absolute value	MOD()	Return the remainder
ACOS()	Return the arc cosine	PI()	Return the value of pi
ASIN()	Return the arc sine	POW()	Return the argument raised to the specified power
ATAN()	Return the arc tangent	POWER()	Return the argument raised to the specified power
ATAN2(), ATA	Return the arc tangent of the two arguments	RADIANS()	Return argument converted to radians
CEIL()	Return the smallest integer value not less than the argument	RAND()	Return a random floating-point value
CEILING()	Return the smallest integer value not less than the argument	ROUND()	Round the argument
CONV()	Convert numbers between different number bases	SIGN()	Return the sign of the argument
COS()	Return the cosine	SIN()	Return the sine of the argument
COT()	Return the cotangent	SQRT()	Return the square root of the argument
CRC32()	Compute a cyclic redundancy check value	TAN()	Return the tangent of the argument
DEGREES()	Convert radians to degrees	TRUNCATE()	Truncate to specified number of decimal places
DIV	Integer division		

2.3. Funciones de fecha.

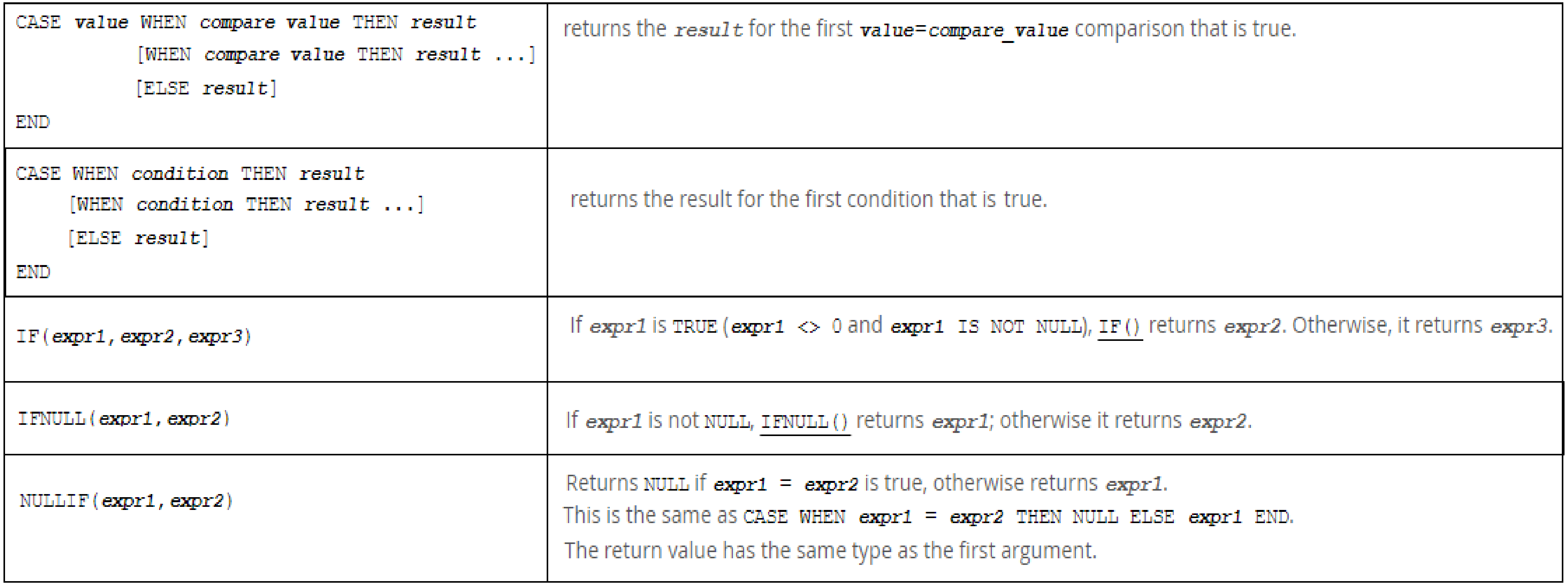
ADDDATE()	Add time values (intervals) to a date value	MONTH()	Return the month from the date passed
ADDTIME()	Add time	MONTHNAME()	Return the name of the month
CONVERT_TZ()	Convert from one time zone to another	NOW()	Return the current date and time
CURDATE()	Return the current date	PERIOD_ADD()	Add a period to a year-month
CURRENT_DATE(), CURRENT_DATE	Synonyms for CURDATE()	PERIOD_DIFF()	Return the number of months between periods
CURRENT_TIME(), CURRENT_TIME	Synonyms for CURTIME()	QUARTER()	Return the quarter from a date argument
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP()	Synonyms for NOW()	SEC_TO_TIME()	Converts seconds to 'hh:mm:ss' format
CURTIME()	Return the current time	SECOND()	Return the second (0-59)
DATE()	Extract the date part of a date or datetime expression	STR_TO_DATE()	Convert a string to a date
DATE_ADD()	Add time values (intervals) to a date value	SUBDATE()	Synonym for DATE_SUB() when invoked with three arguments
DATE_FORMAT()	Format date as specified	SUBTIME()	Subtract times
DATE_SUB()	Subtract a time value (interval) from a date	SYSDATE()	Return the time at which the function executes
DATEDIFF()	Subtract two dates	TIME()	Extract the time portion of the expression passed
DAY()	Synonym for DAYOFMONTH()	TIME_FORMAT()	Format as time
DAYNAME()	Return the name of the weekday	TIME_TO_SEC()	Return the argument converted to seconds
DAYOFMONTH()	Return the day of the month (0-31)	TIMEDIFF()	Subtract time
DAYOFWEEK()	Return the weekday index of the argument	TIMESTAMP()	expression; with two arguments, the sum of the arguments
DAYOFYEAR()	Return the day of the year (1-366)	TIMESTAMPADD()	Add an interval to a datetime expression
EXTRACT()	Extract part of a date	TIMESTAMPDIFF()	Subtract an interval from a datetime expression
FROM_DAYS()	Convert a day number to a date	TO_DAYS()	Return the date argument converted to days
FROM_UNIXTIME()	Format Unix timestamp as a date	TO_SECONDS()	Return the date or datetime argument converted to seconds since Year 0
GET_FORMAT()	Return a date format string	UNIX_TIMESTAMP()	Return a Unix timestamp
HOUR()	Extract the hour	UTC_DATE()	Return the current UTC date
LAST_DAY	Return the last day of the month for the argument	UTC_TIME()	Return the current UTC time
LOCALTIME(), LOCALTIME	Synonym for NOW()	UTC_TIMESTAMP()	Return the current UTC date and time
LOCALTIMESTAMP, LOCALTIMESTAMP()	Synonym for NOW()	WEEK()	Return the week number
MAKEDATE()	Create a date from the year and day of year	WEEKDAY()	Return the weekday index
MAKETIME()	Create time from hour, minute, second	WEEKOFYEAR()	Return the calendar week of the date (1-53)
MICROSECOND()	Return the microseconds from argument	YEAR()	Return the year
MINUTE()	Return the minute from the argument	YEARWEEK()	Return the year and week

2.4. Funciones de agregación.



AVG()	Return the average value of the argument
BIT_AND()	Return bitwise AND
BIT_OR()	Return bitwise OR
BIT_XOR()	Return bitwise XOR
COUNT()	Return a count of the number of rows returned
COUNT(DISTINCT)	Return the count of a number of different values
GROUP_CONCAT()	Return a concatenated string
JSON_ARRAYAGG()	Return result set as a single JSON array
JSON_OBJECTAGG()	Return result set as a single JSON object
MAX()	Return the maximum value
MIN()	Return the minimum value
STD()	Return the population standard deviation
STDDEV()	Return the population standard deviation
STDDEV_POP()	Return the population standard deviation
STDDEV_SAMP()	Return the sample standard deviation
SUM()	Return the sum
VAR_POP()	Return the population standard variance
VAR_SAMP()	Return the sample variance
VARIANCE()	Return the population standard variance

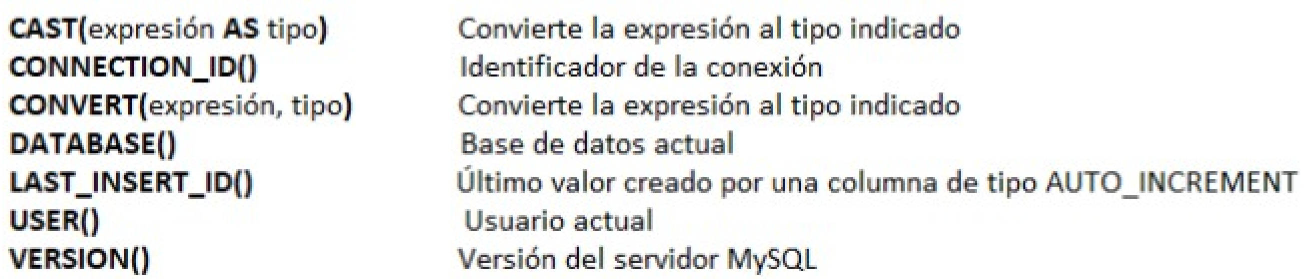
2.5. Funciones de control de flujo.



CASE *value* WHEN *compare value* THEN *result* [WHEN *compare value* THEN *result* ...] [ELSE *result*] END	returns the *result* for the first *value*=*compare_value* comparison that is true.
CASE WHEN *condition* THEN *result* [WHEN *condition* THEN *result* ...] [ELSE *result*] END	returns the result for the first condition that is true.
IF(*expr1*,*expr2*,*expr3*)	If *expr1* is TRUE (*expr1* <> 0 and *expr1* IS NOT NULL), IF() returns *expr2*. Otherwise, it returns *expr3*.
IFNULL(*expr1*,*expr2*)	If *expr1* is not NULL, IFNULL() returns *expr1*; otherwise it returns *expr2*.
NULLIF(*expr1*,*expr2*)	Returns NULL if *expr1* = *expr2* is true, otherwise returns *expr1*. This is the same as CASE WHEN *expr1* = *expr2* THEN NULL ELSE *expr1* END. The return value has the same type as the first argument.

A3. Funciones propias del SGBD.

2.6. Otras funciones.



CAST(expresión **AS** tipo**)**	Convierte la expresión al tipo indicado
CONNECTION_ID()	Identificador de la conexión
CONVERT(expresión, tipo**)**	Convierte la expresión al tipo indicado
DATABASE()	Base de datos actual
LAST_INSERT_ID()	Último valor creado por una columna de tipo AUTO_INCREMENT
USER()	Usuario actual
VERSION()	Versión del servidor MySQL