SQL

SQL KEYWORDS DESCRIPTION

ADD

Adds a new column to an existing table.

Example: Adds a new column named email\_address' to a table named ALTER TABLE

ADD email address It creates a new constraint on an existing table, which is used to specify rules for any data in the table.

Example: Adds

new PRIMARY KEY

ADD CONSTRAINT

constraint named 'user' on columns

ID and SURNAME.

ALTER TABLE users

ADD CONSTRAINT user PRIMARY KEY (ID, SURNAME);

Adds, deletes or edits columns in a table. It can also be used to add and delete constraints in a table, as per the above Example: Adds a new boolean column called approved to a table named 'deals! ALTER TABLE deals

1234567

SELECT \* FROM employees;

3 SELECT \* FROM employees WHERE emp\_id = 2;

6 SELECT \* FROM employees 7 ORDER BY name;

SQL KEYWORDS

KEYWORDS

DESCRIPTION

Deletes a column from a table.

Example: Removes the first\_name column

DROP COLUMN

from the users table.

ALTER TABLE users

DROP COLUMN first\_name

Deletes the entire database.

DROP DATABASE

Example: Deletes a database named 'websitesetup.

DROP DATABASE websitesetup;

DROP DEFAULT

Removes a default value for a column.

Example 1(MySQL): Removes the default value from the name column in the 'products' table.

ALTER TABLE products

ALTER COLUMN name DROP DEFAULT;

Deletes a table from a database

DROP

ALTER TABLE

TABLE

ADD approved boolean;

Example 2: Deletes the approved column

from the 'deals' table.

ALTER TABLE deals

DROP COLUMN approved;

Changes the data type of a table's column.

ALTER COLUMN

Example In the 'users' table, make the column 'incept\_date' into a 'datetime' type. ALTER TABLE users

EXISTS

ALTER COLUMN incept\_date datetime;

Returns true if all of the subquery values

meet the passed condition.

Example: Returns the users with a higher number of tasks than the user with the highest number of tasks in the HR

Example: Removes the users table.

DROP TABLE users;

Checks for the existence of any record within the subquery, returning true one or more records are returned.

Example: Lists any dealerships with a deal. finance percentage less than 10.

SELECT dealership\_name

FROM dealerships

WHERE EXISTS (SELECT deal\_name FROM deals WHERE dealership\_id = deals.dealership\_id AND finance\_

percentage< 10);

Specifies which table to select or delete data from.

ALL

Example: Selects data from the users table.

department (id 2).

SELECT first\_name, surname, tasks\_no FROM users

FROM

WHERE tasks\_no > ALL (SELECT tasks

FROM user WHERE department\_id = 2);

Used to join separate conditions within a WHERE clause.

Example:Returns events located in

London, United Kingdom.

AND

SELECT FROM events

WHERE host\_country='United Kingdom'

IN

AND host\_city="London';

Returns true if any of the subquery values meet the given condition.

Example:Returns products from the

SELECT area\_manager

FROM area\_managers

WHERE EXISTS (SELECT ProductName FROM Products WHERE area\_manager\_id = deals.area\_manager\_id AND Price < 20);

Used alongside a WHERE cause as a shorthand for multiple OR conditions. So instead of:

SELECT FROM users

WHERE country = 'USA' OR country = 'United Kingdom' OR

country 'Russia' OR country = 'Australia';

You can use:-

SELECT FROM users

WHERE country IN (USA', 'United

Kingdom', 'Russia', 'Australia');

ANY

products table which have received orders stored in the orders table - with a quantity of more than 5.

SELECT name

FROM products

WHERE productid= ANY (SELECT productid FROM orders WHERE Qty >

AS

Renames a table or column with an alias value which only exists for the duration of the query.

Example: Aliases north\_east\_user\_ subscriptions column.

SELECT north\_east\_user\_subscriptions AS ne subs

FROM users

WHERE ne\_subs > 5;

INSERT INTO

IS NULL

IS NOT NULL

Add new rows to a table.

Example: Adds a new vehicle.

INSERT INTO cars (make, model, mileage, year) VALUES ('Audi', 'A3', 30000, 2016);

Tests for empty (NULL) values.

Example: Returns users that haven't given

a contact number.

SELECT FROM users

WHERE contact\_number IS NULL;

The reverse of NULL Tests for values that aren't empty / NULL.

Returns true if the operand value matches a pattern.

Example: Returns true if the user's

Used with ORDER BY to return the data in ascending order.

LIKE

first\_name ends with 'son'.

ASC

Example:Apples, Bananas, Peaches, Raddish.

SELECT FROM users

WHERE first\_name LIKE '%son';

Selects values within the given range.

Example 1:Selects stock with a quantity

between 100 and 150.

Retums true if a record DOESN'T meet the condition.

Example: Returns true if the user's

SELECT FROM stock

NOT

first\_name doesn't end with 'son'.

SELECT FROM users

WHERE quantity BETWEEN 100 AND 150;

BETWEEN

Example 2:Selects stock with a quantity NOT between 100 and 150. Alternatively, using the NOT keyword here reverses the logic and selects values outside the given range.

SELECT FROM stock

OR

WHERE quantity NOT BETWEEN 100 AND 150;

WHERE first\_name NOT LIKE 'son';

Used alongside WHERE to include data. when either condition is true.

Example: Returns users that live in either Sheffield or Manchester. SELECT FROM users

WHERE city = 'Sheffield' OR 'Manchester';

CASE

Change query output depending on conditions.

Example 1:Returns users and their subscriptions, along with a new column called activity levels that makes a judgement based on the number of subscriptions.

SELECT first\_name, surname, subscriptions

CASE WHEN subscriptions > 10 THEN 'Very active'

ORDER BY

WHEN Quantity BETWEEN 3 AND 10 THEN 'Active'

ROWNUM

ELSE 'Inactive'

END AS activity levels

FROM users;

Adds a constraint that limits the value which

can be added to a column.

Example 1(MySQL): Makes sure any users added to the users table are 18 or over.

SELECT

CREATE TABLE users

first\_name varchar(255),

age int,

CHECK

CHECK (age=18)

):

Example 2(MySQL): Adds a check after

SELECT DISTINCT

the table has already been created.

ALTER TABLE users

ALTER TABLE users

Creates a new database.

CREATE DATABASE

Example 1(MySQL): Creates a new database named websiteset.

SELECT INTO

CREATE DATABASE websitesetup;

Creates a new table.

Example: Creates a new table called 'users' in the websitesetup database. CREATE TABLE users

SELECT TOP

id int,

CREATE TABLE

first\_name varchar(255),

surname varchar(255),

address varchar(255),

contact number int

);

SET

Used to sort the result data in ascending (default) or descending order through the use of ASC or DESC keywords. Example: Returns countries in alphabetical order.

SELECT FROM countries

ORDER BY name;

Retums results where the row number

meets the passed condition..

Example: Returns the top 10 countries from the countries table.

SELECT \* FROM countries

WHERE ROWNUM C= 10;

Used to select data from a database, which

is then returned in a results set.

Example Selects all columns from all

users.

SELECT FROM users;

Example 2: Selects the first\_name and

surname columns from all users.XX

SELECT first\_name, surname FROM users;

Sames as SELECT, except duplicate values are excluded.

Example Creates a backup table using data from the users table. SELECT INTO usersBackup2020 FROM users;

Copies data from one table and inserts it into another.

Example: Returns all countries from the users table, removing any duplicate values. (which would be highly likely). SELECT DISTINCT country from users;

Allows you to return a set number of

records to retum from a table. Example: Returns the top 3 cars from the cars table.

SELECT TOP 3 FROM cars;

Used alongside UPDATE to update existing data in a table.

Example: Updates the value and quantity values for an order with an id of 642 in the orders table.

UPDATE orders

SET value 19.49, quantity = 2

Sets a default value for a column;

Example 1(MySQL): Creates a new table

WHERE id = 642;

called Products which has a name column

with

default value of 'Placeholder Name'

SOME

Identical to ANY

and an available\_from column with a

default value of today's date.

CREATE TABLE products

id int.

TOP

name varchar(255) DEFAULT 'Placeholder

Name',

DEFAULT

available from date DEFAULT GETDATE()

);

Example 2(MySQL): The same as above,

but editing an existing table.

ALTER TABLE products

TRUNCATE TABLE

ALTER name SET DEFAULT 'Placeholder

Name'

ALTER available from SET DEFAULT

GETDATE();

Delete data from a table.

Example: Removes. user with a user\_id

Used alongside SELECT to retum a set number of records from a table.

Example: Returns the top 5 users from the users table.

SELECT TOP 5 FROM users;

Similar to DROP, but instead of deleting the table and its data, this deletes only the data.

Example: Empties the sessions table, but leaves the table itself intact.

TRUNCATE TABLE sessions;

Combines the results from 2 or more SELECT statements and returns only distinct values.

Example: Returns the cities from the events

SELECT city FROM events

UNION

SELECT city from subscribers;

DELETE

of 674.

UNION

and subscribers tables.

DELETE FROM users WHERE user\_id =

674;

Used with ORDER BY to retum the data in descending order.

DESC

Example: Raddish, Peaches, Bananas,

UNION ALL

Apples.

The same as UNION but includes duplicate values.

COMMENTS

SINGLE LINE COMMENTS

Single line comments start with - Any text after these 2 characters to

the end of the line will be ignored

-- My Select query

SELECT FROM users;

CREATE TABLE users (

id int,

first\_name varchar(255)

);

This constraint ensures all values in a column are unique.

UNIQUE

Example 1 (MySQL): Adds a unique

constraint to the id column when creating a new users table.

CREATE TABLE users

id int NOT NULL,

name varchar(255) NOT NULL,

UNIQUE (id)

):

Example 2 (MySQL): Alters an existing column to add a UNIQUE constraint.

ALTER TABLE users

ADD UNIQUE (id);

Updates existing data in a table.

Example: Updates the mileage and service

Due values for a vehicle with an

UPDATE

UPDATE cars

VALUES

WHERE

SET mileage 23500, serviceDue = 8

WHERE id = 45;

Used alongside the INSERT INTO keyword to add new values to a table.

Example: Adds a new car to the cars table.

SET mileage

23500, serviceDue = 8

INSERT INTO cars (name, model, year)

VALUES ('Ford', 'Fiesta', 2010);

Filters results to only include data which

meets the given condition.

Example: Returns orders with a quantity

of more than 1 item.

SELECT FROM orders

WHERE quantity > 1;

MULTILINE COMMENTS

Multiline comments start with /\* and end with \*/. They stretch across

multiple lines until the closing characters have been found.

1

This is my select query.

It grabs all rows of data from the users table

SELECT FROM users;

This is another select query, which I don't

want to execute yet SELECT FROM tasks;

\*/

MySQL Data Types

STRING DATA TYPES

DATA TYPE

CHAR(SIZE)

DESCRIPTION

Fixed length string which can contain letters, numbers and special characters. The size parameter sets the maximum string length, from 0-255 with a default

of 1.

VARCHAR(SIZE)

Variable length string similar to CHARQ,

but with a maximum string length range from 0 to 65535.

NUMERIC DATA TYPES

DATA TYPE

BIT(SIZE)

DESCRIPTION

A bit-value type with a default of 1. The allowed number of bits in a value is set vial

the size parameter, which can hold values from 1 to 64.

A very small integer with a signed range of -128 to 127, and an unsigned range of 0 to 255. Here, the size parameter specifies the maximum allowed display width, which

TINYINT(SIZE)

is 255.

BOOLEAN

Same as BOOL

BINARY(SIZE)

VARCHAR(SIZE)

TINYBLOB

TINYTEXT

TEXT(size)

Similar to CHARO but stores binary byte strings.

Similar to CHARO but stores binary byte strings.

Holds Binary Large Objects (BLOBs) with

a maxlength of 255 bytes.

Holds a string with a maximum length of

255 characters. Use VARCHAR() instead,

as it's fetched much faster.

Holds a string with a maximum length of 65535 bytes. Again, better to use

VARCHARO

SMALLINT(size)

MEDIUMINT(size)

INT(size)

A small integer with a signed range of -32768 to 32767, and an unsigned range from 0 to 65535. Here, the size parameter specifies the maximum allowed display width, which is 255.

A medium integer with a signed range of -8388608 to 8388607, and an unsigned range from 0 to 16777215. Here, the size parameter specifies the maximum allowed display width, which is 255.

A medium integer with a signed range of -2147483648 to 2147483647, and an unsigned range from 0 to 4294967295. Here, the size parameter specifies the maximum allowed display width, which is 255.

BLOB(size)

MEDIUMBLOB

LONGTEXT

Holds Binary Large Objects (BLOBS) with

a max length of 65535 bytes.

Holds Binary Large Objects (BLOBs) with

a max length of 16,777,215 bytes

Holds a string with a maximum length of 4,294,967,295 characters.

Holds Binary Large Objects (BLOBS) with

LONGBLOB

a max length of 4,294,967,295 bytes

INTEGER(size)

Same as INT.

BIGINT(size)

A medium integer with a signed range of

-9223372036854775808 to 9223372036854775807, and an unsigned

range from 0 to 18446744073709551615. Here, the size parameter specifies the maximum allowed display width, which

is 255.

ENUM(a, b, c,

etc...)

SET

(a, b, c, etc...)

DATA TYPE

A string object that only has one value, which is chosen from a list of values which

you define, up to a maximum of 65535 values. If a value is added which isn't on

this list, it's replaced with a blank value

instead. Think of ENUM being similar to

regard.

HTML radio boxes in this CREATE TABLE tshirts (color ENUM 'red',

'green', 'blue', 'yellow', 'purple'));

A string object that can have 0 or more values, which is chosen from a list of values which you define, up to a

maximum of 64 values. Think of SET

being similar to HTML checkboxes

in this regard

DATE/TIME DATA TYPES

DATE

DESCRIPTION

A simple date in YYYY-MM-DD format, with

a supported range from "1000-01-01' to '9999-12-31.

FLOAT(p)

DOUBLE(size, d)

DECIMAL(size, d)

DEC(sze, d)

A floating point number value. If the

precision (p) parameter is between O to 24, then the data type is set to FLOAT(), whilst if its from 25 to 53, the data type is set to DOUBLE(). This behaviour is to make the storage of values more efficient.

A floating point number value where the total digits are set by the size parameter, and the number of digits after the decimal point is set by the d parameter

An exact fixed point number where the

total

number of digits is set by the size

parameters, and the total number of digits

after the decimal point is set by the d parameter.

For size, the maximum number is 65 and the default is 10, whilst ford, the maximum

number is 30 and the default is 10.

Same as DECIMAL.

A date time in YYYY-MM-DD hh:mmess

format, with a supported range from '1000-01-01 00:00:00' to '9999-12-31

DATETIME(fsp)

23:59:59'.'

By adding DEFAULT and ON UPDATE to

the column definition, it automatically

sets

to the current date/time.

TIMESTAMP(fsp)

A Unix Timestamp, which is a value

relative to the number of seconds since

the Unix epoch (1970-01-01 00:00:00' UTC). This has a supported range from '1970-01-01 00:00:01 UTC to 2038-01-09

03:14:07 UTC

By adding DEFAULT UPDATE CURRENT TIMESTAMP to the

CURRENT\_TIMESTAMP and ON

column definition, it automatically sets to

current date/time.

TIME(fsp)

YEAR

A time in hh:mmess format, with al supported range from '838:59:59 to

'838:59:59.

A year, with a supported range of '1901' to

'2155.

ARITHMETIC OPERATORS

OPERATOR

OPERATORS

BITWISE OPERATORS

DESCRIPTION

DESCRIPTION

OPERATOR

Add

&

Bitwise AND

I

Bitwise OR

Subtract

Multiply

Divide

%

Modulo

COMPARISON OPERATORS

Λ

Bitwise exclusive OR

COMPOUND OPERATORS

OPERATOR

DESCRIPTION

+=

Add equals

OPERATOR

DESCRIPTION

\_=

Subtract equals

Equal to

\*=

Multiply equals

>

Greater than

/=

Divide equals

<

Less than

%=

Modulo equals

>=

Greater than or equal to

&=

Bitwise AND equals

<=

Less than or equal to

^=

Bitwise exclusive equals

Not equal to

|\*=

Bitwise OR equals

FUNCTIONS

STRING FUNCTIONS

DATA TYPE

ASCIIA

CHAR\_LENGTH

CHARACTER

LENGTH

CONCAT

DESCRIPTION

Retums the equivalent ASCII value for a specific character.

Retums the character length of a string.

Same as CHAR\_LENGTH.

NUMERIC FUNCTIONS

DATA TYPE

ABS

DESCRIPTION

Returns the absolute value of the given

number.

ACOS

Returns the arc cosine of the given number,

ASIN

Returns the arc sine of the given number.

Adds expressions together, with a minimum.

of 2.

ATAN

Returns the arc tangent of one or 2 given

numbers

CONCAT\_WS

Adds expressions together, but with a

separator between each value.

ATAN2

Return the arc tangent of 2 given numbers,

FIELD

Retums an index value relative to the position of a value within a list of values.

AVG

Returns the average value of the given

expression.

When passed a number, retums that number.

FORMAT

CEIL

formatted to include commas (eg 3,400,000).

Returns the closest whole number (integer)

upwards from a given decimal point number.

Allows you to insert one string into another

CEILING

Same as CEIL

INSERT

at a certain point, for a certain number of

characters.

COS

Returns the cosine of a given number.

Retums the position of the first time one

INSTR

string appears within another.

COT

Retums the cotangent of a given number.

LCASE

Convert a string to lowercase.

Starting from the left, extract the given

number of characters from a string and

LEFT

return them as another.

LENGTH

Retums the length of a string, but in bytes.

COUNT

DEGREES

DIV

EXP

LOCATE

Retums the first occurrence of one string

within another,

LOWER

LPAD

Same as LCASE

Left pads one string with another, to a

specific length.

Remove any leading spaces from the given.

LTRIM

string.

MID

Extracts one string from another, starting

from any position.

POSITION

REPEAT

REPLACE

REVERSE

Returns the position of the first time one

substring appears within another.

Allows you to repeat a string.

Allows you to replace any instances of a

substring within a string, with a new

substring.

Reverses the string.

Starting from the right, extract the given

Returns the amount of records that are

returned by a SELECT query.

Converts a radians value to degrees.

Allows you to divide integers.

Returns to the power of the given number.

Returns the closest whole number (integer)

downwards from a given decimal point

FLOOR

number.

GREATEST

LEAST

LN

Returns the highest value in a list of

arguments.

Returns the smallest value in a list of

arguments.

Retums the natural logarithm of the given

number.

Returns the natural logarithm of the given

number, or the logarithm of the given number

LOG

to the given base.

LOG10

Does the same as LOG, but to base 10

LOG2

Does the same as LOG, but to base 2.

MAX

Returns the highest value from a set of

values.

RIGHT

number characters from a string and

return them as another.

Returns the lowest value from a set

MIN

of values.

RPAD

Right pads one string with another, to at

specific

length.

Removes any trailing spaces from the given

RTRIM

string.

MOD

Returns the remainder of the given number

divided by the other given number.

PI

Returns Pl.

Returns a string full of spaces equal to the

SPACE

amount you pass it.

Returns the value of

POW

the given number

raised to the power of the other given

number.

STRCMP

SUBSTR

Compares 2 strings for differences.

Extracts one substring from another, starting

from any position.

SUBSTRING

Same as SUBSTR

SUBSTRING

INDEX

TRIM

UCASE

Returns a substring from a string before the

passed substring is found the number of

times equals to the passed number.

Removes trailing and leading spaces from

the given string Same as if you were to run

LTRIM and RTRIM together.

Convert a string to uppercase.

UPPER

Same as UCASE.

NUMERIC FUNCTIONS

DATA TYPE

ADDDATE

DESCRIPTION

Add a date interval (eg 10 DAY) to a date

(eg 20/01/20) and return the result (eg:

20/01/30).

ADDTIME

Add a time interval (eg: 02:00) to a time or

datetime (05:00) and return the result

(07:00).

POWER

RADIANS

Same as POW.

Converts a degrees value to radians.

RAND

Returns random number.

SIGN

SIN

SQRT

SUM

Returns the sign of the given number.

Returns the sine of the given number.

Returns the square root of the given

number.

Returns the value of the given set of values

combined.

TAN

Returns the tangent of the given number.

TRUNCATE

Returns number truncated to the given

number of decimal places.

NUMERIC FUNCTIONS

DATA TYPE

IN

BINARY

DESCRIPTION

Returns the given number in binary.

Returns the given value as a binary string.

CAST

COALESCE

Convert one type into another.

From a list of values, return the first non-null

CURDATE

CURRENT\_DATE

CURRENT

TIMESTAMP

CURTIME

Get the current date.

Same as CURDATE.

Get the current date and time.

Same as CURRENT\_TIME.

CONNECTION\_ID

CONV

CONVERT

value.

For the current connection, return the

unique

connection ID.

Convert the given number from one numeric

base system into another.

Convert the given value into the given

datatype or character set.

DATE

DATEDIFF

Extracts the date from a datetime

expression.

Returns the number of days between the 2

given dates.

CURRENT\_USER

DATABASE

Return the user and hostname which was

used to authenticate with the server.

Get the name of the current database.

DATE\_ADD

DATE\_FORMAT

Same as ADDDATE.

Formats the date to the given pattern.

DATE\_SUB

GROUP BY

Subtract a date interval (eg: 10 DAY) to

a

date (eg 20/01/20) and return the result

(eg:

20/01/10).

DAY

DAYNAME

Returns the day for the given date.

Returns the weekday name for the given

date.

DAYOFWEEK

DAYOFYEAR

Returns the index for the weekday for the

given date.

Returns the day of the year for the given

date

EXTRACT

Extract from the date the given part (eg

MONTH for 20/01/20=01).

FROM DAYS

Return the date from the given numeric date

value. HOUR.

HOUR

Return the hour from the given date.

LAST DAY

HAVING

IF

IFNULL

Used alongside aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group

the results.

Example: Lists the number of users with

active orders.

SELECT COUNT(user\_id), active\_orders

FROM users

GROUP BY active\_orders;

It's used in the place of WHERE with

aggregate

functions.

Example: Lists the number of users with

active orders, but only include users with

more than 3 active orders. SELECT COUNT(user\_id), active\_orders

FROM users

GROUP BY active\_orders

HAVING COUNT(user\_id) > 3;

If the condition is true return a value,

otherwise return another value.

If the expression is null, return 1 otherwise

return O.

Get the last day of the month for the given

date.

LAST\_INSERT\_ID

For the last row which was added or

updated in a table, return the auto increment.

LOCALTIME

Gets the current local date and time.

LOCALTIMESTAMP Same as LOCALTIME.

NULLIF

Compares the 2 given expressions. If they

are equal, NULL is returned, otherwise the

first expression is returned.

MAKEDATE

MICROSECOND

MINUTE

Creates a date and returns it, based on the given year and number of days values.

Returns the microsecond of a given time or

datetime.

Returns the minute of the given time or

datetime.

SESSION\_USER

SYSTEM\_USER

VERSION

Return the current user and hostnames.

Same as SESSION\_USER.

Returns the current version of the MySQL

powering the database..

MONTH

MONTHNAME

Returns the month of the given date.

Returns the name of the month of the given

date

NOW

Same as LOCALTIME.

PERIOD\_ADD

Adds the given number of months to the

given period.

PERIOD\_DIFF

Returns the difference between 2 given.

periods.

QUARTER

SECOND

Returns the year quarter for the given date.

Returns the second of a given time or

datetime.

NAME

%

WILDCARDS

DESCRIPTION

Equates to zero or more characters.

Example 1: Find all users with surnames

ending in 'son'.

SELECT FROM users

WHERE surname LIKE 'xson';

Example 2: Find all users with surnames

ending in 'son'.

SELECT \* FROM users

WHERE city LIKE '%che %';

SEC\_TO\_TIME

STR\_TO\_DATE

SUBDATE

SUBTIME

SYSDATE

TIME

Returns time based on the given seconds.

Creates a date and returns it based on the

given string and format.

Same as DATE\_SUB.

Subtracts a time interval (eg: 02:00) to a time or datetime (05:00) and return the

result (03:00).

Same as LOCALTIME.

Returns the time from a given time or

datetime.

TIME\_FORMAT

TIME\_TO\_SEC

Returns the given time in the given format.

Converts and returns a time into seconds.

(CHADIST)

Equates to any single character.

Example 1: Find all users living in cities beginning with any 3 characters, followedby

'chester!

SELECT FROM users

WHERE city LIKE chester'

Equates to any

single character in the list.

Example 1: Find all users with first names

beginning with J, H or M.

SELECT FROM users

WHERE first\_name LIKE '(jhm]>';

Example 2: Find all users with first names

beginning letters between A-L

SELECT FROM users

WHERE first\_name LIKE '[a-1]>';

Example 3: Find all users with first names

not ending with letters between n-s. SELECT \* FROM users

WHERE first\_name LIKE '%[!n-s]);

TIMEDIFF

TIMESTAMP

Returns the difference between 2 given

time/datetime expressions.

Returns the datetime value of the given

date or datetime.

TO\_DAYS

Returns the total number of days that have

passed from '00-00-0000' to the given date.

WEEK

WEEKDAY

Returns the week number for the given date.

Returns the weekday number for the given

date.

WEEKOFYEAR

YEAR

Returns the week number for the given date.

Returns the year from the given date.

YEARWEEK

Returns the year and week number for the

given date.

KEYS

FOREIGN KEY

USERS

EXAMPLE 1 (MYSQL)

CREATE TABLE orders (

id

Candidate Key

int

id int NOT NULL,

first\_name

varchar

user\_id int,

product\_id int,

PRIMARY KEY (id),

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (product\_id) REFERENCES products(id)

ORDERS

last\_name

varchar

id

int

address

varchar

user\_id (Foreign Key) int

):

email

varchar

product\_id (Foreign Key) int

agent\_logged

Parent Table

tinyint

Child Table

EXAMPLE 1 (MYSQL)

CREATE TABLE users (

id int NOT NULL AUTO\_INCREMENT,

first\_name varchar(255).

last\_name varchar(255) NOT NULL,

address varchar(255).

email varchar(255).

PRIMARY KEY (id)

);

PRODUCTS

id Candidate Key

int

name

varchar

description

text

stock\_count

int

price

float

Parent Table

PRIMARY KEY

EXAMPLE 2 (MYSQL)

ALTER TABLE orders

ADD FOREIGN KEY (user\_id) REFERENCES users(id);

EXAMPLE 2 (MYSQL)

ALTER TABLE users

ADD PRIMARY KEY (first\_name);

DATA TYPE

CREATE

INDEX

WILDCARDS

DESCRIPTION

Creates an index named 'idx\_test' on the

first\_name and

surname columns of the

users table. In this instance, duplicate

values

are allowed.

CREATE INDEX idx\_test

ON users (first\_name, surname);

CREATING VIEWS

VIEWS

CREATE VIEW priority\_users AS

SELECT FROM users

WHERE country United Kingdom;

CREATE

UNIQUE

INDEX

Creates an index named "idx\_test' on the

first name and surname columns of the

users table. In this instance, duplicate

values are allowed.

CREATE UNIQUE INDEX idx\_test

ON users (first\_name, surname);

SELECT FROM [priority\_users);

DROP

Creates an index named "idx test' on the

first name and surname columns of the

users table. In this instance, duplicate

REPLACING VIEWS

CREATE OR REPLACE VIEW [priority\_users] AS

SELECT FROM users

WHERE country = 'United Kingdom' OR country="USA";

INDEX

values are allowed.

ALTER TABLE users

DROP INDEX idx\_test;

JOINS

INNER JOIN

LEFT JOIN

DELETING VIEWS

DROP VIEW priority\_users;

ORDERS

address

id

first\_name Last\_name

1

Lube

Harison

1640, Kjetil Homme..

email

Lube@153....

2

Healer

Reynolds

742, Norway, Denmark. heal132@hot...

Table 1

Table 2

Table 1

Table 2

3

Simpson

Chekson

7, Nova Scotia.....

4

Chekson

Simpson

15, Santo Domingo....

5

Oliver

Harison

1640, San Salvador.

RIGHT JOIN

OUTER JOIN

6

jones

Gabet

598, Caracas, 1010

Simpson76@....

Checkson80@....

oliver5715@....

jones547@....

7

Micheal

Johnson

12, Western Michigan..

Micheal0017@....

Table 1

Table 2

Table 1

Table 2

8

Thomes

Smith

342, Mary Jones Station smith098@....

9

Robyn

Gabet

598, Monte, Trigo..

robyn65478@....

10

Byony

Brown

165, First Ave. Usa..

byony8754@....

ORDERS

PROUDCTS

id

user\_id

product\_id

agent\_logged

id

name

description

stock\_count

price

1

5

196

0

102

Cartono.

Why

0

14,99

2

4

32

1

23

Cardbor..

Dec

1

3.49

3

6

310

0

3

Smart..

NULL

1

24.99

4

10

196

1

32

Troast 33.

Die

4

09.50

5

1

67

1

275

A4..

berta

5

4.99

6

1

341

1

436

Pack of 50.

you

5

12.99

1

875

0

341

Set of 25.

Modo

2

4.99

8

9

3

1

67

Large Car..

Pack

10

12.99

9

5

23

1

196

10, XP..

Hay que

10

15.99

10

8

196

1

310

Set of 35

Cettal

10

2.99

PROUDCTS

SELECT orders.id, users.first\_name, users.surname, products.name as

'product name

id

name description

stock\_count

price

FROM orders

102

Cartono.

Why

0

14,99

INNER JOIN users

on orders.user\_id = users.id

23

Cardbor..

Dec

1

3.49

3

Smart.

NULL

1

24.99

32

Troast 33.

Die

4

09.50

275

A4..

berta

5

4.99

INNER JOIN products on orders.product\_id = products.id;

INNER JOIN RESULT SET

id first\_name surname product name

436

Pack of 50.

you

5

12.99

1

Oliver

Harison

10 X Plastic....

341

Set of 25..

Modo

2

4.99

2

Claire

Simpson

TripLast 33...

67

Large Car..

Pack

10

12.99

3

James

Gilbert

StorePac 5...

196

10, XP..

Hay que

10

15.99

4

Bryony

Brown

10 X Plasti....

310

Set of 35

Cetta

10

2.99

5

Luke

Harison

Large Care....

6

Luke

Harison

Set of 2 S..

7

Luke

Harison

A4 Storage...

8

Robyn

Gilbert

SmartMo..

9

Oliver

Harrison

Cardboar...

10

Thomas

Smith

10 X Plasti

SQL KEYWORDS

KEYWORDS

ADD

ADD CONSTRAINT

DESCRIPTION

Adds a new column to an existing table.

Example: Adds a new column named

email address' to a table named 'users!-

ALTER TABLE users

ADD email address varchar(255);

It creates a new constraint on an existing

table, which is used to specify rules for any

data in the table.

Example: Adds a new PRIMARY KEY

constraint named 'user' on columns

ID and SURNAME

ALTER TABLE users

ADD CONSTRAINT user PRIMARY KEY

(ID, SURNAME);