Kanak

SQL

```
SELECT * FROM table;
SELECT * FROM table
WHERE condition;
SELECT * FROM table
ORDER BY column;
```

SQL KEYWORDS

KEYWORDS	DESCRIPTION
ADD	Adds a column in an existing table, or adds a constraint on a table (some systems support this keyword only for adding constraints)
ADD CONSTRAINT	Adds a constraint after a table is already created, so you should be careful to specify a constraint name for this one
ALTER TABLE	Adds, deletes, or modifies columns in a table, or changes the data type of columns in a table. Also used to add and drop various constraints on an existing table.
ALTER COLUMN	Changes the data type of a column in a table. Also used to add and drop default constraints on an existing table.
ALL	Returns true if all of the subqueries listed return true. Must be preceded by =, !=, >, <, <=, or >=
AND	Logical operator. Returns true if all of the subexpressions connected by AND are true
ANY	Returns true if any of the subqueries listed return true. Must be preceded by =, !=, >, <, <=, or >=
AS	Renames a column or table with an alternate name (an alias) only for the duration of the query
ASC	Specifies that the results should be sorted in ascending order. Values sort from lowest to highest
AVG	Returns the average value in a numeric column
BETWEEN	Specifies a range to test within, for example, "BETWEEN 10 AND 20"
CASE	Begins a conditional expression, followed by one or more WHEN ... THEN statements
CHECK	Specifies a constraint that limits the values that can be placed in a column. For instance, "CHECK (salary > 0)"
CREATE CONSTRAINT	Creates a constraint after a table is already created
CREATE INDEX	Creates an index on a table. Indexes are used to retrieve data more quickly than otherwise
CREATE VIEW	Creates a view, based on one or more table queries
DATABASE	Creates a database
DEFAULT	Specifies a default value for a column in a table, for use when no value is specified when adding a new row to the table
DELETE	Deletes one or more records from a table
DESC	Specifies that the results should be sorted in descending order. Values sort from highest to lowest
DISTINCT	Eliminates duplicate results from a query
DROP	Deletes an entire table, view, or database
DROP COLUMN	Deletes a column in a table
DROP CONSTRAINT	Deletes a constraint on a table
DROP DATABASE	Deletes an database
DROP INDEX	Deletes an index on a table
DROP VIEW	Deletes a view
EXISTS	Returns true if a subquery contains any rows
FROM	Specifies the table from which to select or delete records
GROUP BY	Groups results by a column or columns
HAVING	Specifies a condition for groups created by a GROUP BY statement
IN	Returns true if a specified value matches any of a list of values
INDEX	Specifies an index to be created on a table
INNER JOIN	Returns rows from two or more tables when there is a match between columns in the tables
INSERT INTO	Inserts new records into a table
IS NULL	Returns true if a specified value is null
IS NOT NULL	Returns true if a specified value is not null
LIKE	Compares a specified value to part of a column using wildcards
LIMIT	Specifies the maximum number of records to return in a query
NOT	Logical operator. Returns true if the following condition is false
OR	Logical operator. Returns true if any of the following conditions are true
ORDER BY	Specifies the order in which to return the results of a query
OUTER JOIN	Returns rows from two or more tables when there is a match between columns in the tables, and also returns unmatched rows
PRIMARY KEY	Specifies a column or combination of columns that will be used to uniquely identify each row in a table
RIGHT JOIN	Returns rows from two or more tables when there is a match between columns in the tables, and also returns unmatched rows from the table on the right side of the JOIN
ROWNUM	Returns a specified number of records from the top of a table
SELECT	Retrieves records from one or more tables in a database
SELECT DISTINCT	Returns only unique values in a column
SELECT TOP	Returns a specified number of records from the top of a table
SET	Specifies which values to update in a table
SOME	Returns true if any of the following subqueries return true
TOP	Specifies the number of records to return in a query
TRUNCATE TABLE	Deletes all records in a table, but not the table itself
UNION	Combines the results of two or more queries into a single result set, omitting any duplicate rows
UNION ALL	Combines the results of two or more queries into a single result set, including all duplicate rows
UNIQUE	Specifies that all values in a column must be different
UPDATE	Updates one or more records in a table. Usually used with a WHERE clause to specify which records to update
VALUES	Specifies the values to be inserted into a table. For example, "INSERT INTO table (column1, column2) VALUES (value1, value2)"
WHERE	Specifies a condition for records to be returned by a query. For example, "WHERE salary > 30000"

SQL KEYWORDS

KEYWORDS	DESCRIPTION
ADD	Adds a column in an existing table, or adds a constraint on a table (some systems support this keyword only for adding constraints)
ADD CONSTRAINT	Adds a constraint after a table is already created, so you should be careful to specify a constraint name for this one
ALTER TABLE	Adds, deletes, or modifies columns in a table, or changes the data type of columns in a table. Also used to add and drop various constraints on an existing table.
ALTER COLUMN	Changes the data type of a column in a table. Also used to add and drop default constraints on an existing table.
ALL	Returns true if all of the subqueries listed return true. Must be preceded by =, !=, >, <, <=, or >=
AND	Logical operator. Returns true if all of the subexpressions connected by AND are true
ANY	Returns true if any of the subqueries listed return true. Must be preceded by =, !=, >, <, <=, or >=
AS	Renames a column or table with an alternate name (an alias) only for the duration of the query
ASC	Specifies that the results should be sorted in ascending order. Values sort from lowest to highest
AVG	Returns the average value in a numeric column
BETWEEN	Specifies a range to test within, for example, "BETWEEN 10 AND 20"
CASE	Begins a conditional expression, followed by one or more WHEN ... THEN statements
CHECK	Specifies a constraint that limits the values that can be placed in a column. For instance, "CHECK (salary > 0)"
CREATE CONSTRAINT	Creates a constraint after a table is already created
CREATE INDEX	Creates an index on a table. Indexes are used to retrieve data more quickly than otherwise
CREATE VIEW	Creates a view, based on one or more table queries
DATABASE	Creates a database
DEFAULT	Specifies a default value for a column in a table, for use when no value is specified when adding a new row to the table
DELETE	Deletes one or more records from a table
DESC	Specifies that the results should be sorted in descending order. Values sort from highest to lowest
DISTINCT	Eliminates duplicate results from a query
DROP	Deletes an entire table, view, or database
DROP COLUMN	Deletes a column in a table
DROP CONSTRAINT	Deletes a constraint on a table
DROP DATABASE	Deletes an database
DROP INDEX	Deletes an index on a table
DROP VIEW	Deletes a view
EXISTS	Returns true if a subquery contains any rows
FROM	Specifies the table from which to select or delete records
GROUP BY	Groups results by a column or columns
HAVING	Specifies a condition for groups created by a GROUP BY statement
IN	Returns true if a specified value matches any of a list of values
INDEX	Specifies an index to be created on a table
INNER JOIN	Returns rows from two or more tables when there is a match between columns in the tables
INSERT INTO	Inserts new records into a table
IS NULL	Returns true if a specified value is null
IS NOT NULL	Returns true if a specified value is not null
LIKE	Compares a specified value to part of a column using wildcards
LIMIT	Specifies the maximum number of records to return in a query
NOT	Logical operator. Returns true if the following condition is false
OR	Logical operator. Returns true if any of the following conditions are true
ORDER BY	Specifies the order in which to return the results of a query
OUTER JOIN	Returns rows from two or more tables when there is a match between columns in the tables, and also returns unmatched rows
PRIMARY KEY	Specifies a column or combination of columns that will be used to uniquely identify each row in a table
RIGHT JOIN	Returns rows from two or more tables when there is a match between columns in the tables, and also returns unmatched rows from the table on the right side of the JOIN
ROWNUM	Returns a specified number of records from the top of a table
SELECT	Retrieves records from one or more tables in a database
SELECT DISTINCT	Returns only unique values in a column
SELECT TOP	Returns a specified number of records from the top of a table
SET	Specifies which values to update in a table
SOME	Returns true if any of the following subqueries return true
TOP	Specifies the number of records to return in a query
TRUNCATE TABLE	Deletes all records in a table, but not the table itself
UNION	Combines the results of two or more queries into a single result set, omitting any duplicate rows
UNION ALL	Combines the results of two or more queries into a single result set, including all duplicate rows
UNIQUE	Specifies that all values in a column must be different
UPDATE	Updates one or more records in a table. Usually used with a WHERE clause to specify which records to update
VALUES	Specifies the values to be inserted into a table. For example, "INSERT INTO table (column1, column2) VALUES (value1, value2)"
WHERE	Specifies a condition for records to be returned by a query. For example, "WHERE salary > 30000"

Kanak Infosystems LLP.

COMMENTS

SINGLE LINE COMMENTS

```
-- This is a single line comment
SELECT * FROM table;
```

MULTILINE COMMENTS

```
/* This is a multiline comment
which can span many lines
SELECT * FROM table;
*/
```

MySQL Data Types

(version 8.0)

String types
Numeric types
Date and time types

STRING DATA TYPES

DATA TYPE	DESCRIPTION
CHAR(SIZE)	A fixed length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(SIZE)	A variable length string (can contain letters, numbers, and special characters). The size parameter specifies the maximum column length in characters - can be from 0 to 65535
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(SIZE)	Holds a string with a maximum length of 65535 bytes
MEDIUMTEXT	Holds a string with a maximum length of 16777215 characters
LONGTEXT	Holds a string with a maximum length of 4294967295 characters
ENUM(val1, val2, val3, ...)	A string object that can have only one value, chosen from a list of possible values. You can enter up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can enter up to 64 values in a SET list
BINARY(SIZE)	Equal to CHAR(), but stores binary byte strings. The size parameter specifies the column length in bytes. Default is 1
VARBINARY(SIZE)	Equal to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large Objects). Holds up to 255 bytes
BLOB(SIZE)	Holds a BLOB (Binary Large Objects) value. The size parameter specifies the maximum length in bytes, 0 - 65535
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16777215 bytes
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4294967295 bytes

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
BIT(SIZE)	A bit-value type. The number of bits per value is specified in size. The size parameter can hold a value from 1 bit to 64 bits. The default value for size is 1.
TINYINT(SIZE)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The size parameter specifies the maximum display width (which is 255)
SMALLINT(SIZE)	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The size parameter specifies the maximum display width (which is 255)
MEDIUMINT(SIZE)	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The size parameter specifies the maximum display width (which is 255)
INT(SIZE)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The size parameter specifies the maximum display width (which is 255)
BIGINT(SIZE)	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The size parameter specifies the maximum display width (which is 255)
FLOAT(p)	A floating point number. MySQL uses a single precision floating point number in MySQL. Where the parameter p is from 0 to 24 the default data type is FLOAT() or FLOAT(24)
DOUBLE(size, d)	A floating point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter
DECIMAL(size, d), NUMERIC(size, d)	An exact fixed point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter

DATE AND TIME DATA TYPES

DATA TYPE	DESCRIPTION
DATE	Date. Format: YYYY-MM-DD
DATETIME(fsp)	Date and time combined. Format: YYYY-MM-DD hh:mm:ss[.fractional seconds]
TIMESTAMP(fsp)	Stores the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss[.fractional seconds]
TIME(fsp)	Time. Format: hh:mm:ss[.fractional seconds]
YEAR	A year in four-digit format (1901 to 2155)

Kanak Infosystems LLP.

OPERATORS

ARITHMETIC OPERATORS

OPERATOR	DESCRIPTION
+	Addition
-	Subtract
*	Multiply
/	Divide
%	Modulo

BITWISE OPERATORS

OPERATOR	DESCRIPTION
|	Bitwise OR
&	Bitwise AND
^	Bitwise XOR

COMPARISON OPERATORS

OPERATOR	DESCRIPTION
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal to
!=	Not equal to
!<	Not less than
!>	Not greater than

COMPOUND OPERATORS

OPERATOR	DESCRIPTION
+=	Add equal
-=	Subtract equal
*=	Multiply equal
/=	Divide equal
%=	Modulo equal
&=	Bitwise AND equal
^-=	Bitwise XOR equal
|*=	Bitwise OR equal

LOGICAL OPERATORS

OPERATOR	DESCRIPTION
ALL	True if all of the subqueries values meet the condition
AND	True if all the conditions separated by AND is TRUE
ANY	True if any of the subqueries values meet the condition
BETWEEN	True if the operand is within the range of comparisons
EXISTS	True if the subquery returns one or more records
IN	True if the operand is equal to one of a list of expressions
LIKE	True if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	True if any of the conditions separated by OR is TRUE
SOME	True if any of the subqueries values meet the condition

Kanak Infosystems LLP.

FUNCTIONS

STRING FUNCTIONS

DATA TYPE	DESCRIPTION
ASCII(x)	Returns the ASCII value for a specified character
CHAR_LENGTH(x)	Returns the length of a string (in characters)
CHARACTER_LENGTH(x)	Returns the length of a string (in characters)
CONCAT(x,y,...)	Adds two or more expressions together
CONCAT_WS(x,y,...)	Adds two or more expressions together with a separator
FIELD(x,y,...)	Returns the index position of a value in a list of values
FIND_IN_SET(x,y)	Returns the first occurrence of a specified string
FORMAT(x,y)	Formats a specified number to a specified format
INSERT(x,y,z,w)	Inserts a string into a specified position in a specified string
INSTR(x,y)	Returns the first position of the first occurrence of a specified string
LCASE(x)	Converts a specified string into lower case
LEFT(x,y)	Returns a specified number of characters from the left side of a specified string
LENGTH(x)	Returns the length of the string, value in bytes
LOCATE(x,y,z)	Returns the position of the first occurrence of a specified string in a specified string
LOWER(x)	Converts a string into lower case
LPAD(x,y,z)	Adds a specified string to the left side of a specified string, a specified number of times
LTRIM(x)	Removes leading spaces from a specified string
MID(x,y,z)	Extracts a specified number of characters from a specified string, starting from a specified position
POSITION(x IN y)	Returns the position of the first occurrence of a specified string in a specified string
REPEAT(x,y)	Repeats a specified string a specified number of times
REPLACE(x,y,z)	Replaces all occurrences of a specified string in a specified string, with another specified string
REVERSE(x)	Reverses a specified string
RIGHT(x,y)	Returns a specified number of characters from the right side of a specified string
RPAD(x,y,z)	Adds a specified string to the right side of a specified string, a specified number of times
RTRIM(x)	Removes trailing spaces from a specified string
SPACE(x)	Returns a specified number of spaces
STRCMP(x,y)	Compares two strings
SUBSTR(x,y,z), SUBSTRING(x,y,z), SUBSTRING_INDEX(x,y,z)	Returns a specified number of characters from a specified string, starting from a specified position
TRIM(x)	Removes leading and trailing spaces from a specified string
UCASE(x)	Converts a specified string into upper case
UPPER(x)	Converts a specified string into upper case

NUMERIC FUNCTIONS

DATA TYPE	DESCRIPTION
ABS(x)	Returns the absolute value of a specified number
ACOS(x)	Returns the arccosine of a specified number
ASIN(x)	Returns the arcsine of a specified number
ATAN(x)	Returns the arctan of a specified number
ATAN2(x,y)	Returns the arctan2 of a specified pair of numbers
CEIL(x)	Returns the smallest integer value that is bigger than or equal to a specified number
CEILING(x)	Returns the smallest integer value that is bigger than or equal to a specified number
COS(x)	Returns the cosine of a specified number
COT(x)	Returns the cotangent of a specified number
CRC32(x)	Returns the 32-bit cyclic redundancy check value of a specified string
DEGREES(x)	Converts a specified number from radians to degrees
DIV	Divisor x by a specified number
EXP(x)	Returns e raised to the power of a specified number
FLOOR(x)	Returns the largest integer value that is smaller than or equal to a specified number
GREATEST(x,y,...)	Returns the greatest value of the specified list of arguments
LN(x)	Returns the natural logarithm of a specified number
LOG(x)	Returns the natural log of a specified number, or the logarithm of a specified number in a specified base
LOG10(x)	Returns the base-10 log of a specified number
LOG2(x)	Returns the base-2 log of a specified number
MOD(x,y)	Returns the remainder of a specified number divided by another specified number
PI()	Returns PI
POW(x,y)	Returns x raised to the power of y
POWER(x,y)	Returns x raised to the power of y
RADIANS(x)	Converts a specified number from degrees to radians
RAND()	Returns a random number
ROUND(x,y)	Rounds a specified number to a specified number of decimal places
SIGN(x)	Returns the sign of a specified number, whether it is negative, positive or zero
SIN(x)	Returns the sine of a specified number
SQRT(x)	Returns the square root of a specified number
TAN(x)	Returns the tangent of a specified number
TRUNCATE(x,y)	Returns a specified number truncated to a specified number of decimal places

NUMERIC FUNCTIONS

DATA TYPE	DESCRIPTION
ADDDATE(x,y)	Returns the date after a specified time/date
ADDTIME(x,y)	Returns a specified time/date, added with a specified time
CURDATE()	Returns the current date
CURRENT_DATE()	Returns the current date
CURRENT_TIME()	Returns the current time
CURRENT_TIMESTAMP()	Returns the current date and time
CURTIME()	Returns the current time
DATE(x)	Returns the date part from a specified date/time
DATEDIFF(x,y)	Returns the number of days between two specified date/time
DATE_FORMAT(x,y)	Formats a specified date/time according to a specified format
DAY(x)	Returns the day part of a specified date/time
DAYNAME(x)	Returns the name of the weekday for a specified date/time
DAYOFMONTH(x)	Returns the day part of a specified date/time
DAYOFWEEK(x)	Returns the weekday number for a specified date/time
DAYOFYEAR(x)	Returns the day of the year for a specified date/time
FROM_DAYS(x)	Returns a date from a specified numeric value
HOUR(x)	Returns the hour part of a specified date/time
LAST_DAY(x)	Returns the last day of the month for a specified date/time
LOCALTIME()	Returns the current date and time
LOCALTIMESTAMP()	Returns the current date and time
MICROSECOND(x)	Returns the microsecond part of a specified date/time
MINUTE(x)	Returns the minute part of a specified date/time
MONTH(x)	Returns the month part of a specified date/time
MONTHNAME(x)	Returns the name of the month for a specified date/time
NOW()	Returns the current date and time
PERIOD_ADD(x,y)	Returns a specified time period added with a specified number of months
PERIOD_DIFF(x,y)	Returns the difference between two specified time periods
QUARTER(x)	Returns the quarter of the year for a specified date/time
SEC_TO_TIME(x)	Returns a specified number of seconds in time format
STR_TO_DATE(x,y)	Returns a specified date/time based on a specified format
SUBDATE(x,y)	Returns a specified date after a specified time/date
SUBTIME(x,y)	Returns a specified time/date, subtracted by a specified time
SYSDATE()	Returns the current date and time
TIME(x)	Returns the time part from a specified date/time
TIME_FORMAT(x,y)	Formats a specified date/time according to a specified format
TIME_TO_SEC(x)	Returns the number of seconds from a specified time
TIMEDIFF(x,y)	Returns the difference between two specified date/time
TIMESTAMP(x,y)	Returns a specified date/time, added with a specified time
TO_DAYS(x)	Returns the number of days from a specified date/time
WEEK(x)	Returns the week number for the given date
WEEKDAY(x)	Returns the weekday number for the given date
WEEKOFYEAR(x)	Returns the week number for the given date
YEAR(x)	Returns the year part of a specified date/time
YEARWEEK(x)	Returns the year and week number for a specified date/time

Kanak Infosystems LLP.

KEYS

FOREIGN KEY

```
EXAMPLE (MYSQL):
CREATE TABLE orders
(order_id int(10),
fog_name varchar(255),
fog_price decimal(8,2),
PRIMARY KEY (order_id),
FOREIGN KEY (fog_name)
REFERENCES fogs(fog_name)
);
```

PRIMARY KEY

```
EXAMPLE (MYSQL):
CREATE TABLE fogs
(fog_name varchar(255),
fog_type varchar(255),
fog_price decimal(8,2),
PRIMARY KEY (fog_name)
);
```

Kanak Infosystems LLP.

DATA TYPES

DATA TYPE	DESCRIPTION
CREATE TABLE	Creates a table in a database
DATE	Stores a date in a format of YYYY-MM-DD
DATETIME	Stores a date and time in a format of YYYY-MM-DD HH:MI:SS
DROP TABLE	Deletes a table in a database
GRANT	Gives a user access privileges to a database
INT	Stores an integer (a whole number without decimals)
KEY	Stands for either PRIMARY KEY or FOREIGN KEY
NULL	Indicates that a field may have no value
NUMERIC	Stores numeric data
REVOKE	Takes away a user's access privileges to a database
TEXT	Stores a string with a maximum length of 65535 characters
TIMESTAMP	Stores a timestamp in a format of YYYY-MM-DD HH:MI:SS
VARCHAR	Stores a variable length character string

VIEWS

CREATING VIEWS

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

REPLACING VIEWS

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

DELETING VIEWS

```
DROP VIEW view_name;
```

JOINS

ORDERS

id	user_id	product_id	quantity	order_date
1	1	1	1	2017-01-01
2	1	2	1	2017-01-01
3	2	1	1	2017-01-01
4	2	2	1	2017-01-01

PRODUCTS

id	name	description	date_added	price
1	Product 1	Description 1	2017-01-01	10.00
2	Product 2	Description 2	2017-01-02	20.00
3	Product 3	Description 3	2017-01-03	30.00

CROSS

id	first_name	last_name	address	email
1	Peter	Griffin	31 Spooner Street, Quahog	griffin@gmail.com
2	Lois	Griffin	31 Spooner Street, Quahog	griffin@gmail.com
3	Joe	Swanson	31 Spooner Street, Quahog	swanson@gmail.com
4	Cleveland	Brown	31 Spooner Street, Quahog	brown@gmail.com

id	name	date_added	price
1	Product 1	2017-01-01	10.00
2	Product 2	2017-01-02	20.00
3	Product 3	2017-01-03	30.00

SELECT * FROM CROSS JOIN table;
SELECT * FROM table1 CROSS JOIN table2;

Kanak Infosystems LLP.

www.kanakinfosystems.com