Kanak

SQL

```
SELECT * FROM [table]
WHERE [condition]
ORDER BY [column]

SELECT * FROM [table]
GROUP BY [column]
HAVING [condition]
```

SQL KEYWORDS

KEYWORDS	DESCRIPTION
ADD	Keyword for adding a column in an existing table
ADD CONSTRAINT	Keyword used to add constraints in an existing table, such as a primary key or a foreign key
ALTER TABLE	Keyword used to add, delete, or modify columns in an existing table. Also used to add and drop various constraints on an existing table
ALTER COLUMN	Keyword used to add, delete, or modify columns in an existing table
AND	Logical operator. Displays a record if both the first condition and the second condition are true
ANY	Returns true if any of the subquery values meet the condition
AS	Keyword used to create an alias for a table or column
ASC	Keyword used to sort the result-set in ascending order
BETWEEN	Keyword used with the WHERE keyword to specify a range of values
CASE	Keyword used to create a conditional statement
CHECK	Keyword used to create a constraint that limits the value that can be placed in a column
CREATE DATABASE	Keyword used to create a database
CREATE INDEX	Keyword used to create an index in a table
DELETE	Keyword used to delete a record in a table
DESC	Keyword used to sort the result-set in descending order
DISTINCT	Keyword used to return only distinct (different) values
DROP	Keyword used to delete a column, a table, or a database
DROP COLUMN	Keyword used to delete a column in a table
EXEC	Keyword used to execute a stored procedure
EXISTS	Keyword used to check if a record exists in a subquery
FOREIGN KEY	Keyword used to create a foreign key
FROM	Keyword used to specify which table to select, delete, or modify records from
GROUP BY	Keyword used to group the result-set by one or more columns
HAVING	Keyword used with the GROUP BY keyword to specify a condition for the result-set
IN	Keyword used to specify multiple values in a WHERE clause
INDEX	Keyword used to create or delete an index in a table
INNER JOIN	Keyword used to join two tables. Returns records that have matching values in both tables
INSERT INTO	Keyword used to insert new records in a table
IS NULL	Keyword used to test for empty values
IS NOT NULL	Keyword used to test for non-empty values
JOIN	Keyword used to join two tables. Returns records that have matching values in both tables
LEFT JOIN	Keyword used to join two tables. Returns all records from the first table, and the records from the second table that have matching values
LIKE	Keyword used in a WHERE clause to search for a specified pattern in a column
NOT	Keyword used with other keywords to negate a condition
NOT NULL	Keyword used to specify that a column should not accept NULL values
OR	Logical operator. Displays a record if either the first condition or the second condition is true
ORDER BY	Keyword used to sort the result-set by one or more columns
OUTER JOIN	Keyword used to join two tables. Returns all records from both tables
PRIMARY KEY	Keyword used to create a primary key
RIGHT JOIN	Keyword used to join two tables. Returns all records from the second table, and the records from the first table that have matching values
SELECT	Keyword used to select records from a database
SELECT DISTINCT	Keyword used to select only distinct values
SELECT INTO	Keyword used to create a new table from an existing table
SELECT TOP	Keyword used to specify the number of records to return
SET	Keyword used to specify which columns and values that are updated in a table
SOME	Returns true if some of the subquery values meet the condition
TOP	Keyword used to specify the number of records to return
TRUNCATE TABLE	Keyword used to delete the data inside a table, but not the table itself
UNION	Operator used to combine the result-set of two or more SELECT statements
UNION ALL	Operator used to combine the result-set of two or more SELECT statements (allows duplicate values)
UNIQUE	Keyword used to create a constraint that ensures that all values in a column are unique
UPDATE	Keyword used to update existing records in a table
VALUES	Keyword used in the INSERT INTO keyword to specify the values to be inserted into a table
WHERE	Keyword used to specify a condition when fetching data from a table, or when updating, inserting into, or deleting from a table

Kanak Infosystems LLP.

COMMENTS

SINGLE LINE COMMENTS

Single line comments start with two dashes (--) and ends at the end of the line.

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

MULTILINE COMMENTS

Multiline comments begin with a slash followed by an asterisk (/*) and ends with an asterisk followed by a slash (*/). Comments can be on one or more lines.

```
/* This is a multiline
comment */
SELECT * FROM table;
/* This is a multiline
comment */
SELECT * FROM table;
```

MySQL Data Types

Click on a data type for more information.

- String data types
- Numeric data types
- Date and time data types

STRING DATA TYPES

DATA TYPE	DESCRIPTION
CHAR(SIZE)	Holds a fixed length character string value from 0 to 255 characters long
VARCHAR(SIZE)	Holds a variable length character string value from 0 to 65535 characters long
BINARY(SIZE)	Holds a fixed length binary string
VARBINARY(SIZE)	Holds a variable length binary string
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(SIZE)	Holds a string with a maximum length of 65535 characters
MEDIUMTEXT	Holds a string with a maximum length of 16777215 characters
LONGTEXT	Holds a string with a maximum length of 4294967295 characters
ENUM(x,y,z,etc.)	Let you enter 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 x,y,z,etc. (x,y,z,etc.)	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
BIT(SIZE)	A bit-value type. Number of bits per value specified in SIZE. The possible values for SIZE are from 1 bit to 64 bits
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)
BOOL or BOOLEAN	Zero is considered as false, non-zero values are considered as true
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)
INTEGER(SIZE)	This type is a synonym for INT.
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)	Used to specify a floating point number. MySQL uses the p value to determine whether to use FLOAT or DOUBLE for the resulting data type. If p is from 0 to 24, the data type becomes FLOAT. If p is from 25 to 53, the data type becomes DOUBLE
DOUBLE(size, d)	A normal size floating point number. The size parameter specifies the total number of digits. The d parameter specifies the number of digits after the decimal point. Floating point numbers are stored as approximations
DECIMAL(size, d), NUMERIC(size, d)	Stores fixed point numbers. The size parameter specifies the total number of digits. The d parameter specifies the number of digits after the decimal point. Fixed point numbers are stored as exact values
DOUBLE(size, d)	Same as DOUBLE

DATE AND TIME DATA TYPES

DATA TYPE	DESCRIPTION
DATE	Stores a date in the format YYYY-MM-DD
DATETIME(fsp)	Stores a date and time in the format YYYY-MM-DD hh:mm:ss. Adding optional fractional seconds precision (fsp) specifies up to six digits after the decimal point
TIMESTAMP(fsp)	Stores a timestamp. Number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Adding optional fractional seconds precision (fsp) specifies up to six digits after the decimal point. Automatically updated when a record is inserted or updated
TIME(fsp)	Stores a time in the format hh:mm:ss. Adding optional fractional seconds precision (fsp) specifies up to six digits after the decimal point
YEAR	Stores a year in a four-digit format (1901 to 2155)

Kanak Infosystems LLP.

OPERATORS

ARITHMETIC OPERATORS

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

COMPARISON OPERATORS

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

BITWISE OPERATORS

OPERATOR	DESCRIPTION
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise not
>>	Bitwise shift right
<<	Bitwise shift left

COMPOUND OPERATORS

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

Kanak Infosystems LLP.

FUNCTIONS

STRING FUNCTIONS

DATA TYPE	DESCRIPTION
ASCII(x)	Returns the ASCII value for the specified character x
CHAR_LENGTH(x)	Returns the length of the string x in characters
CHARACTER_LENGTH(x)	Returns the length of the string x 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 the value x in the list of values y,...
FIND_IN_SET(x,y)	Returns the position of a value x in a list of values y,...
FORMAT(x,y)	Formats the number x to a format like "#,###,###.##", rounded to y decimal places
INSERT(x,y,z,w)	Inserts a string w into string x, beginning at position y, and z characters long
INSTR(x,y)	Returns the first position of the string y in the string x
LCASE(x)	Converts the string x to lower-case
LEFT(x,y)	Returns the left-most y characters from the string x
LENGTH(x)	Returns the length of the string x in bytes
LOCATE(x,y,z)	Returns the position of the first occurrence of the string x in the string y, after position z
LOWER(x)	Converts the string x to lower-case
LPAD(x,y,z)	Adds a string z to the left of string x, such that the total length of string x is y characters long
LTRIM(x)	Removes leading spaces from the string x
MID(x,y,z)	Returns z characters from the string x, beginning at position y
POSITION(x IN y)	Returns the position of the first occurrence of the string x in the string y
REPEAT(x,y)	Repeats the string x y times
REPLACE(x,y,z)	Replaces all occurrences of the string y with the string z in the string x
REVERSE(x)	Reverses the string x
RIGHT(x,y)	Returns the right-most y characters from the string x
RPAD(x,y,z)	Adds a string z to the right of string x, such that the total length of string x is y characters long
RTRIM(x)	Removes trailing spaces from the string x
SPACE(x)	Returns x number of spaces
STRCMP(x,y)	Compares string x to string y
SUBSTR(x,y,z)	Returns z characters from the string x, beginning at position y
SUBSTRING(x,y,z)	Returns z characters from the string x, beginning at position y
SUBSTRING_INDEX(x,y,z)	Returns the string x before z occurrences of the string y
TRIM(x)	Removes leading and trailing spaces from the string x
UCASE(x)	Converts the string x to upper-case
UPPER(x)	Converts the string x to upper-case

NUMERIC FUNCTIONS

DATA TYPE	DESCRIPTION
ABS(x)	Returns the absolute value of the given number x
ACOS(x)	Returns the arccosine of x
ASIN(x)	Returns the arcsine of x
ATAN(x)	Returns the arctan of x
ATAN2(x,y)	Returns the arctan2 of x and y
AVG(x)	Returns the average value of the given expression x
CEIL(x)	Returns the smallest integer value that is bigger than or equal to the given number x
CEILING(x)	Returns the smallest integer value that is bigger than or equal to the given number x
COS(x)	Returns the cosine of the given number x
COT(x)	Returns the cotangent of the given number x
COUNT(x)	Returns the number of records in a table
DEGREES(x)	Converts a value in radians to degrees
DIV	Used for integer division
EXP(x)	Returns e raised to the power of x
FLOOR(x)	Returns the largest integer value that is smaller than or equal to the given number x
GREATEST(x,y,...)	Returns the greatest value of the given list of expressions
LN(x)	Returns the natural logarithm of the given number x
LOG(x)	Returns the base-10 logarithm of the given number x
LOG10(x)	Returns the base-10 logarithm of the given number x
LOG2(x)	Returns the base-2 logarithm of the given number x
MAX(x)	Returns the maximum value in a set of values x
MIN(x)	Returns the minimum value in a set of values x
MOD(x,y)	Returns the modulo value of x divided by y
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 degree value x into radians
RAND()	Returns a random number
ROUND(x,y)	Rounds the given number x to y decimal places
SIGN(x)	Returns the sign of the given number x
SIN(x)	Returns the sine of the given number x
SQRT(x)	Returns the square root of x
STD(x)	Returns the standard deviation of x
STDDEV(x)	Returns the standard deviation of x
SUM(x)	Returns the sum of a set of values x
TAN(x)	Returns the tangent of the given number x
TRUNCATE(x,y)	Truncates x to y decimal places

DATE AND TIME FUNCTIONS

DATA TYPE	DESCRIPTION
ADDDATE(x,y)	Returns the date x after the time interval y has been added
ADDTIME(x,y)	Returns the time x after the time interval y has been added
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 the date/datetime x
DATEDIFF(x,y)	Returns the difference between the date part of x and the date part of y
DATE_FORMAT(x,y)	Formats the date x according to the format specified in y
DAY(x)	Returns the day part of the given date x
DAYNAME(x)	Returns the name of the weekday for the given date x
DAYOFMONTH(x)	Returns the day of the month for the given date x
DAYOFWEEK(x)	Returns the weekday index for the given date x
DAYOFYEAR(x)	Returns the day of the year for the given date x
FROM_DAYS(x)	Returns the date that is the given number of days from year 0000
HOUR(x)	Returns the hour part of the given date x
LAST_DAY(x)	Returns the last day of the month for the given date x
LOCALTIME()	Returns the current date and time
LOCALTIMESTAMP()	Returns the current date and time
MICROSECOND(x)	Returns the microsecond part of the given date x
MINUTE(x)	Returns the minute part of the given date x
MONTH(x)	Returns the month part of the given date x
MONTHNAME(x)	Returns the name of the month for the given date x
NOW()	Returns the current date and time
PERIOD_ADD(x,y)	Returns the date that is the given number of months from year 0000
PERIOD_DIFF(x,y)	Returns the difference between two given months
QUARTER(x)	Returns the quarter for the given date x
SECOND(x)	Returns the second part of the given date x
SEC_TO_TIME(x)	Returns the time in hh:mm:ss format for the given number of seconds
STR_TO_DATE(x,y)	Returns a date based on the given string x and a format specified in y
SUBDATE(x,y)	Returns the date x after the time interval y has been subtracted
SUBTIME(x,y)	Returns the time x after the time interval y has been subtracted
SYSDATE()	Returns the current date and time
TIME(x)	Returns the time part from the given date/datetime x
TIMEDIFF(x,y)	Returns the difference between two given time/datetime expressions
TIMESTAMP(x)	Returns the datetime value for the given date x
TIMESTAMPADD(x,y,z)	Returns the datetime value of the given date/datetime z plus the interval y
TIMESTAMPDIFF(x,y,z)	Returns the difference between two given date/datetime expressions
TIME_FORMAT(x,y)	Formats the time x according to the format specified in y
TIME_TO_SEC(x)	Returns the time in seconds for the given time x
WEEK(x)	Returns the week number for the given date x
WEEKDAY(x)	Returns the weekday number for the given date x
WEEKOFYEAR(x)	Returns the week number for the given date x
YEAR(x)	Returns the year part of the given date x
YEARWEEK(x)	Returns the year and week number for the given date x

Kanak Infosystems LLP.

KEYS

FOREIGN KEY

```
EXAMPLE (MYSQL):
CREATE TABLE users (
user_id INT,
name VARCHAR(255),
email VARCHAR(255),
PRIMARY KEY (user_id)
);

CREATE TABLE orders (
order_id INT,
user_id INT,
order_date DATE,
PRIMARY KEY (order_id),
FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

PRIMARY KEY

```
EXAMPLE (MYSQL):
CREATE TABLE users (
user_id INT,
name VARCHAR(255),
email VARCHAR(255),
PRIMARY KEY (user_id)
);
```

Kanak Infosystems LLP.

DATABASE

DATA TYPE	DESCRIPTION
CREATE DATABASE [name]	Creates a database. Optional: CHARACTER SET [name] COLLATE [name]
ALTER DATABASE [name]	Changes the character set or collation for a database. Optional: CHARACTER SET [name] COLLATE [name]
DROP DATABASE [name]	Deletes a database and all its tables and views. Optional: IF EXISTS
USE [name]	Selects a database to use for subsequent queries
SHOW DATABASES	Lists all databases on the server
SHOW CREATE DATABASE [name]	Shows the CREATE statement used to create a database

TABLES

DATA TYPE	DESCRIPTION
CREATE TABLE [name] (...)	Creates a table. Optional: IF NOT EXISTS, TEMPORARY, ENGINE=..., DEFAULT CHARSET=...
ALTER TABLE [name] ADD [col] [type]	Adds a column. Optional: AFTER [col], FIRST
ALTER TABLE [name] DROP [col]	Drops a column
ALTER TABLE [name] MODIFY [col] [type]	Changes a column's type
ALTER TABLE [name] CHANGE [old] [new] [type]	Renames and changes a column
ALTER TABLE [name] RENAME TO [new]	Renames the table
DROP TABLE [name]	Deletes a table. Optional: IF EXISTS
TRUNCATE TABLE [name]	Deletes all rows but keeps the table structure
SHOW TABLES	Lists all tables in the current database
SHOW CREATE TABLE [name]	Shows the CREATE statement used to create a table
DESCRIBE [name]	Shows column names, types, nullability, keys, defaults

INDEXES

DATA TYPE	DESCRIPTION
CREATE INDEX [name] ON [table] ([col])	Creates an index. Optional: UNIQUE, IF NOT EXISTS
DROP INDEX [name] ON [table]	Deletes an index. Optional: IF EXISTS
SHOW INDEX FROM [table]	Lists all indexes on a table

VIEWS

DATA TYPE	DESCRIPTION
CREATE VIEW [name] AS [select]	Creates a virtual table based on a query. Optional: OR REPLACE, ALGORITHM=..., DEFINER=..., SQL SECURITY ..., WITH CHECK OPTION
ALTER VIEW [name] AS [select]	Changes the query of an existing view
DROP VIEW [name]	Deletes a view. Optional: IF EXISTS
SHOW CREATE VIEW [name]	Shows the CREATE statement used to create a view

JOINS

ORDERS

id	user_id	product_id	quantity	order_date
1	101	201	2	2024-01-15
2	102	202	1	2024-01-16
3	103	203	3	2024-01-17

PRODUCTS

id	name	category	price	stock
201	Laptop	Electronics	1200.00	50
202	Mouse	Electronics	25.00	200
203	Desk	Furniture	350.00	75

CRUDS

id	first_name	last_name	email	phone	created_at
1	John	Doe	jdoe@mail.com	555-0101	2024-01-15
2	Jane	Smith	jsmith@mail.com	555-0102	2024-01-16
3	Bob	Wilson	bwilson@mail.com	555-0103	2024-01-17

Kanak Infosystems LLP.