Kanak

SQL

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

SQL KEYWORDS

KEYWORDS	DESCRIPTION
ADD	Adds a column in an existing table, or adds a constraint on a column in an existing table
ADD CONSTRAINT	Adds a constraint after a table is already created, so that the first column can be a foreign key
ALTER TABLE	Adds, deletes, or modifies columns in a table, or changes the data type of columns in a table. Also used to add various constraints on existing tables
ALTER COLUMN	Changes the data type of a column in a table. Also used to add or drop default constraints on columns
ALL	Returns true if all of the subqueries meet the condition. Used with SELECT, WHERE and HAVING statements
AND	Used in a WHERE or HAVING clause to specify more than one condition. Also used in JOINS
ANY	Returns true if any of the subqueries meet the condition. Used with SELECT, WHERE and HAVING statements
AS	Used to create an alias for a table or column in a table. Also used to rename a table in a query
ASC	Used in an ORDER BY clause to sort the result set in ascending order
BETWEEN	Used in a WHERE clause to specify a range of values. Also used in JOINS
CASE	Used in a SELECT statement to return a value when a specified condition is met. Also used in UPDATE statements
CHECK	Used in a CREATE TABLE statement to specify a condition that must be met for all values in a column. Also used in ALTER TABLE statements
CREATE DATABASE	Used to create a database. Also used in DROP DATABASE and ALTER DATABASE statements
CREATE TABLE	Used to create a table in a database. Also used in DROP TABLE statements
DEFAULT	Used in a CREATE TABLE statement to specify a default value for a column. Also used in ALTER TABLE statements
DELETE	Used to delete rows in a table. Also used in JOINS
DESC	Used in an ORDER BY clause to sort the result set in descending order
DISTINCT	Used in a SELECT statement to return only distinct values. Also used in JOINS
DROP	Used to delete a database, table, or column in a table. Also used in ALTER statements
DROP COLUMN	Used to delete a column in a table. Also used in ALTER TABLE statements
DROP CONSTRAINT	Used to delete a constraint after a table is created. Also used in ALTER TABLE statements
DROP DATABASE	Used to delete a database. Also used in CREATE DATABASE statements
DROP DEFAULT	Used to delete a default value for a column. Also used in ALTER TABLE statements
EXISTS	Returns true if a subquery contains any records. Also used in JOINS
FROM	Used in a SELECT statement to specify the table from which to retrieve data. Also used in DELETE statements
FULL OUTER JOIN	Returns all records when there is a match in either left or right table. Also used in JOINS
GROUP BY	Used in a SELECT statement to group the result set by one or more columns
HAVING	Used in a SELECT statement to specify a condition for a group or aggregate. Also used in JOINS
IN	Used in a WHERE clause to specify multiple values. Also used in JOINS
INDEX	Used to create or delete a table index
INNER JOIN	Returns records that have matching values in both tables. Also used in JOINS
INSERT INTO	Used to insert new records in a table. Also used in JOINS
IS NULL	Used in a WHERE clause to specify a NULL value. Also used in JOINS
IS NOT NULL	Used in a WHERE clause to specify a non-NULL value. Also used in JOINS
JOIN	Used to join rows from two or more tables, based on a related column between them. Also used in JOINS
LEFT JOIN	Returns all records from the left table, and the matched records from the right table. Also used in JOINS
LIKE	Used in a WHERE clause to search for a specified pattern in a column. Also used in JOINS
NOT	Used in a WHERE clause to specify a condition that is not true. Also used in JOINS
OR	Used in a WHERE clause to specify more than one condition. Also used in JOINS
ORDER BY	Used in a SELECT statement to sort the result set in ascending or descending order
OUTER JOIN	Returns all records when there is a match in either left or right table. Also used in JOINS
PRIMARY KEY	Used in a CREATE TABLE statement to specify a primary key. Also used in ALTER TABLE statements
RIGHT JOIN	Returns all records from the right table, and the matched records from the left table. Also used in JOINS
ROWNUM	Used in a SELECT statement to specify the maximum number of records to return. Also used in JOINS
SELECT	Used to select data from a database. Also used in JOINS
SELECT DISTINCT	Used in a SELECT statement to return only distinct values. Also used in JOINS
SELECT INTO	Used to create a new table from an existing table. Also used in JOINS
SELECT TOP	Used in a SELECT statement to specify a fixed number of records to return. Also used in JOINS
SET	Used in an UPDATE statement to specify which columns and values that should be updated in a table. Also used in JOINS
SOME	Returns true if any of the subqueries meet the condition. Used with SELECT, WHERE and HAVING statements
TOP	Used in a SELECT statement to specify a fixed number of records to return. Also used in JOINS
TRUNCATE TABLE	Used to delete the data inside a table, but not the table itself. Also used in JOINS
UNION	Combines the result-set of two or more SELECT statements. Also used in JOINS
UNION ALL	Combines the result-set of two or more SELECT statements (allows duplicate values). Also used in JOINS
UNIQUE	Used in a CREATE TABLE statement to specify a unique constraint for a column. Also used in ALTER TABLE statements
UPDATE	Used in a SELECT statement to update existing records in a table. Also used in JOINS
VALUES	Used in an INSERT INTO statement to specify the values to be inserted into a table. Also used in JOINS
WHERE	Used to specify a condition when fetching data from a table. Also used in JOINS

Kanak Infosystems LLP.

COMMENTS

SINGLE LINE COMMENTS

Single line comments start with two dashes (--) and end 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 end with an asterisk followed by a slash (*/). Comments can be on one line or many lines.

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

MySQL Data Types

(click to view larger)

MySQL data types

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
TINYINT(size)	-128 to 127 signed range. 0 to 255 unsigned range. Optional specification of display width (size)
SMALLINT(size)	-32768 to 32767 signed range. 0 to 65535 unsigned range. Optional specification of display width (size)
MEDIUMINT(size)	-8388608 to 8388607 signed range. 0 to 16777215 unsigned range. Optional specification of display width (size)
INT(size)	-2147483648 to 2147483647 signed range. 0 to 4294967295 unsigned range. Optional specification of display width (size)
BIGINT(size)	-9223372036854775808 to 9223372036854775807 signed range. 0 to 18446744073709551615 unsigned range. Optional specification of display width (size)
FLOAT(p)	Floating point number. MySQL uses p to determine whether to use FLOAT or DOUBLE
DOUBLE(size, d)	Floating point number. The number of digits after the decimal point is specified in the d parameter
DECIMAL(size, d), DEC(size, d)	An exact fixed-point number. The number of digits after the decimal point is specified in the d parameter
BIT(size)	A bit-value type. The number of bits per value is specified in size. Default is 1

STRING DATA TYPES

DATA TYPE	DESCRIPTION
CHAR(size)	Fixed length character string. Specify length in parenthesis. Default is 1
VARCHAR(size)	Variable length character string. Specify maximum length in parenthesis
BINARY(size)	Equal to CHAR(), but stores binary byte strings. Specify length in parenthesis. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. Specify maximum length in parenthesis
TINYBLOB	For BLOBs (Binary Large Objects). Holds 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65535 bytes of data
TEXT(size)	Holds a string with a maximum length of 65535 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16777215 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16777215 characters
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4294967295 bytes of data
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.)	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

DATE AND TIME DATA TYPES

DATA TYPE	DESCRIPTION
DATE	Date in format YYYY-MM-DD
DATETIME(fsp)	Date and time in format YYYY-MM-DD HH:MM:SS. Optional specification of fractional seconds precision (fsp)
TIMESTAMP(fsp)	Stores the number of seconds since the Unix epoch (1970-01-01 00:00:00 UTC). Optional specification of fractional seconds precision (fsp)
TIME(fsp)	Time in format HH:MM:SS. Optional specification of fractional seconds precision (fsp)
YEAR	A year in format yyyy or yy. Default is 4-digit format

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 left
>>	Bitwise shift right

COMPOUND OPERATORS

OPERATOR	DESCRIPTION
+=	Add equal
-=	Subtract equal
*=	Multiply equal
/=	Divide equal
%=	Modulo equal
&=	Bitwise AND equal
^-=	Bitwise exclusive 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, measured 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 value x followed by a comma-separated list of arguments
FIND_IN_SET(x,y)	Returns the position of a value x in a list of values y, separated by commas
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 for z characters
INSTR(x,y)	Returns the position of the first occurrence of string y in string x
LCASE(x)	Converts a string x into lower case
LEFT(x,y)	Returns the left y characters of 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 string x in string y, after position z
LOWER(x)	Converts a string x into 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 string x
MID(x,y,z)	Returns a substring z characters long from string x, beginning at position y
POSITION(x IN y)	Returns the position of the first occurrence of string x in string y
REPEAT(x,y)	Repeats string x y times
REPLACE(x,y,z)	Replaces all occurrences of string y in string x with string z
REVERSE(x)	Reverses string x and returns the result
RIGHT(x,y)	Returns the right y characters of 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 string x
SPACE(x)	Returns x number of spaces
STRCMP(x,y)	Compares string x to string y
SUBSTR(x,y,z), SUBSTRING(x,y,z)	Returns a substring z characters long from string x, beginning at position y
SUBSTRING_INDEX(x,y,z)	Returns a substring from string x before z occurrences of string y
TRIM(x)	Removes leading and trailing spaces from string x
UCASE(x)	Converts a string x into upper case
UPPER(x)	Converts a string x into 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, where x is a number between -1 and 1
ASIN(x)	Returns the arcsine of x, where x is a number between -1 and 1
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 greater than or equal to the given number x
CEIL(x)	Returns the smallest integer value that is greater than or equal to the given number x
COS(x)	Returns the cosine of x
COT(x)	Returns the cotangent of x
COUNT(x)	Returns the number of records in a table
DIV	Used for integer division
EXP(x)	Returns e raised to the power of x
FLOOR(x)	Returns the largest integer value that is less than or equal to the given number x
GREATEST(x,y,...)	Returns the greatest value in the list of expressions
LN(x)	Returns the natural logarithm of x
LOG(x)	Returns the natural logarithm of x
LOG10(x)	Returns the base-10 logarithm of x
LOG2(x)	Returns the base-2 logarithm of 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 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 value x in degrees to radians
RAND()	Returns a random number
ROUND(x,y)	Rounds a number x to y decimal places
SIGN(x)	Returns the sign of the given number x
SIN(x)	Returns the sine of x, where x is in radians
SQRT(x)	Returns the square root of x
SUM(x)	Returns the total sum of a set of values x
TAN(x)	Returns the tangent of the given number x in radians
TRUNCATE(x,y)	Returns x truncated to y decimal places

DATE FUNCTIONS

DATA TYPE	DESCRIPTION
ADDDATE(x,y)	Returns the date x and a specified time interval y added to it
ADDTIME(x,y)	Returns the time x and a specified time interval y added to it
CURDATE()	Returns the current date
CURRENT_DATE()	Returns the current date
CURRENT_TIME()	Returns the current time
CURTIME()	Returns the current time
DATE()	Returns the date in a specified date or datetime expression
DATE_FORMAT(x,y)	Formats the date x according to the specified format y
DATE_SUB(x,y)	Subtracts a specified time interval y from date x
DAY(x)	Returns the day of the month from date x
DAYNAME(x)	Returns the name of the weekday for date x
DAYOFMONTH(x)	Returns the day of the month from date x
DAYOFWEEK(x)	Returns the weekday index for date x
DAYOFYEAR(x)	Returns the day of the year from date x
FROM_DAYS(x)	Returns a date from the given numeric value x
HOUR(x)	Returns the hour from date x
LAST_DAY(x)	Returns the last day of the month for the given date x
LOCALTIME()	Returns the current date and time
LOCALTIME()	Returns the current date and time
MAKEDATE(x,y)	Creates a date from the given year x and a number of days y
MAKETIME(x,y,z)	Creates a time from the given hour x, minute y, and second z
MICROSECOND(x)	Returns the microsecond from the given time x
MINUTE(x)	Returns the minute from the given time x
MONTH(x)	Returns the month from date x
NOW()	Returns the current date and time
PERIOD_ADD(x,y)	Returns a period x added with y months
PERIOD_DIFF(x,y)	Returns the difference between two periods x and y
QUARTER(x)	Returns the quarter for the given date x
SECOND(x)	Returns the second from date x
SEC_TO_TIME(x)	Returns a time value from the given second x
STR_TO_DATE(x,y)	Returns a date from a string x according to a specified format y
SUBDATE(x,y)	Subtracts a specified time interval y from date x
SUBTIME(x,y)	Subtracts a specified time interval y from time x
SYSDATE()	Returns the current date and time
TIME()	Returns the time from the given date or datetime expression
TIME_FORMAT(x,y)	Formats the time x according to the specified format y
TIME_TO_SEC(x)	Returns the number of seconds from the given time x
TIMEDIFF(x,y)	Returns the difference between two time/datetime expressions x and y
TIMESTAMP(x,y)	Returns a datetime value from a date or datetime expression x and a time expression y
TO_DAYS(x)	Returns the number of days from the given date x
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 from date x
YEARWEEK(x)	Returns the year and week number for the given date

Kanak Infosystems LLP.

KEYS

FOREIGN KEY

```
EXAMPLE (MYSQL):
CREATE TABLE orders (
id int,
product_id int,
user_id int,
FOREIGN KEY (id)
REFERENCES
products(id)
);
```

```
EXAMPLE (MYSQL):
CREATE TABLE users (
id int,
name varchar(255),
email varchar(255),
PRIMARY KEY (id)
);
```

Kanak Infosystems LLP.

PRIMARY KEY

```
EXAMPLE (MYSQL):
CREATE TABLE users (
id int,
name varchar(255),
PRIMARY KEY (id)
);
```

```
EXAMPLE (MYSQL):
ALTER TABLE users
ADD PRIMARY KEY (id);
```

Kanak Infosystems LLP.

DATABASE

DATA TYPE	DESCRIPTION
CREATE DATABASE name	Creates a database with the specified name
ALTER DATABASE name	Changes the database character set
CREATE DATABASE IF NOT EXISTS name	Creates a database with the specified name if it does not exist
DROP DATABASE name	Deletes a database with the specified name
DROP DATABASE IF EXISTS name	Deletes a database with the specified name if it exists

JOINS

ORDERS

id	user_id	product_id	quantity	created
1	1	1	1	2019-01-01
2	1	2	2	2019-01-02
3	2	1	1	2019-01-03

PRODUCTS

id	name	category	price
1	Product 1	Category 1	10.00
2	Product 2	Category 1	20.00
3	Product 3	Category 2	30.00

USERS

id	name	email	age
1	John Doe	john@example.com	25
2	Jane Smith	jane@example.com	30
3	Bob Johnson	bob@example.com	35

CRUDS

id	first_name	last_name	address	email
1	Anna	Smith	Main St	anna@mail.com
2	Antonio	Banderas	Broadway	antonio@mail.com
3	Mary	Jane	Seventh Avenue	mary@mail.com
4	Chris	Evans	Chestnut St	chris@mail.com
5	Linda	Perez	Packard Ln	linda@mail.com

SHOW (MYSQL ONLY)

id	first_name	lastname	phone number
1	Anna	Smith	555-1234
2	Antonio	Banderas	555-5678
3	Mary	Jane	555-9012
4	Chris	Evans	555-3456
5	Linda	Perez	555-7890

SELECT column1, column2, column3, column4, column5 FROM table;

SELECT * FROM table;

SELECT * FROM table WHERE id = 1;

SELECT * FROM table WHERE id = 1 AND name = 'Anna';

Kanak Infosystems LLP.

TABLES

DATA TYPE	DESCRIPTION
CREATE TABLE name	Creates a table with the specified name
DROP TABLE name	Deletes a table with the specified name
DROP TABLE IF EXISTS name	Deletes a table with the specified name if it exists
TRUNCATE TABLE name	Deletes the data inside a table, but not the table itself
ALTER TABLE name ADD column_name datatype	Adds a column in a table
ALTER TABLE name DROP COLUMN column_name	Deletes a column in a table
ALTER TABLE name RENAME TO new_name	Changes the name of a table
ALTER TABLE name RENAME COLUMN old_name TO new_name	Changes the name of a column in a table
ALTER TABLE name ALTER COLUMN column_name datatype	Changes the data type of a column in a table
ALTER TABLE name ADD CONSTRAINT constraint_name constraint_type (column_name)	Adds a constraint after a table is created
ALTER TABLE name DROP CONSTRAINT constraint_name	Deletes a constraint after a table is created
CREATE TABLE new_table AS SELECT column1, column2, column3 FROM table;	Creates a table from an existing table
CREATE TABLE IF NOT EXISTS name	Creates a table with the specified name if it does not exist

CREATING VIEWS

```
CREATE VIEW view_name AS
SELECT column1, column2, column3
FROM table;
```

REPLACING VIEWS

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, column3
FROM table;
```

DELETING VIEWS

```
DROP VIEW view_name;
```

Kanak Infosystems LLP.

INDEXES

DATA TYPE	DESCRIPTION
CREATE INDEX name ON table (column)	Creates an index on a table
CREATE UNIQUE INDEX name ON table (column)	Creates a unique index on a table
DROP INDEX name ON table	Deletes an index on a table
DROP INDEX name	Deletes an index on a table

NULL VALUES

```
SELECT column1, column2, column3
FROM table
WHERE column1 IS NULL;
```

```
SELECT column1, column2, column3
FROM table
WHERE column1 IS NOT NULL;
```

Kanak Infosystems LLP.

AUTO INCREMENT

```
CREATE TABLE table (
column1 datatype AUTO_INCREMENT,
column2 datatype,
column3 datatype,
);
```

```
INSERT INTO table (column2, column3)
VALUES (value2, value3);
```

```
UPDATE table
SET column2 = value2, column3 = value3
WHERE column1 = value1;
```

```
DELETE FROM table
WHERE column1 = value1;
```

```
SELECT * FROM table
ORDER BY column1 ASC, column2 DESC
LIMIT 5;
```

```
SELECT * FROM table
WHERE column1 = value1
AND column2 = value2
OR column3 = value3;
```

```
SELECT * FROM table
WHERE column1 IN (value1, value2, value3);
```

```
SELECT * FROM table
WHERE column1 BETWEEN value1 AND value2;
```

```
SELECT * FROM table
WHERE column1 LIKE 'pattern';
```

```
SELECT * FROM table
WHERE column1 IS NULL;
```

```
SELECT * FROM table
WHERE column1 IS NOT NULL;
```

Kanak Infosystems LLP.

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
BIT(size)	A bit-value type. The number of bits per value is specified in size. Default is 1. Valid range is 1 - 64
TINYINT(size)	-128 to 127 signed range. 0 to 255 unsigned range. Optional specification of display width (size)
BOOL	Zero is considered as false, non-zero values are considered as true
BOOLEAN	Equal to BOOL
SMALLINT(size)	-32768 to 32767 signed range. 0 to 65535 unsigned range. Optional specification of display width (size)
MEDIUMINT(size)	-8388608 to 8388607 signed range. 0 to 16777215 unsigned range. Optional specification of display width (size)
INT(size)	-2147483648 to 2147483647 signed range. 0 to 4294967295 unsigned range. Optional specification of display width (size)
INTEGER(size)	Equal to INT(size)
BIGINT(size)	-9223372036854775808 to 9223372036854775807 signed range. 0 to 18446744073709551615 unsigned range. Optional specification of display width (size)
FLOAT(size, d)	Floating point number. Specify maximum digits in size. Specify maximum digits after the decimal point in d
FLOAT(p)	Floating point number. MySQL uses p to determine whether to use FLOAT or DOUBLE
DOUBLE(size, d)	Floating point number. Specify maximum digits in size. Specify maximum digits after the decimal point in d
DOUBLE PRECISION(size, d)	Floating point number. Specify maximum digits in size. Specify maximum digits after the decimal point in d
DECIMAL(size, d)	An exact fixed-point number. Specify maximum digits in size. Specify maximum digits after the decimal point in d
DEC(size, d)	Equal to DECIMAL(size,d)

DATE AND TIME DATA TYPES

DATA TYPE	DESCRIPTION
DATE	Date in format YYYY-MM-DD
DATETIME(fsp)	Date and time in format YYYY-MM-DD HH:MM:SS. Optional specification of fractional seconds precision (fsp)
TIMESTAMP(fsp)	Stores the number of seconds since the Unix epoch (1970-01-01 00:00:00 UTC). Optional specification of fractional seconds precision (fsp)
TIME(fsp)	Time in format HH:MM:SS. Optional specification of fractional seconds precision (fsp)
YEAR	Year in format yyyy. Default is 4-digit format

Kanak Infosystems LLP.

STRING DATA TYPES

DATA TYPE	DESCRIPTION
CHAR(size)	Fixed length character string. Specify length in parenthesis. Default is 1
VARCHAR(size)	Variable length character string. Specify maximum length in parenthesis
BINARY(size)	Equal to CHAR(), but stores binary byte strings. Specify length in parenthesis. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. Specify maximum length in parenthesis
TINYBLOB	For BLOBs (Binary Large Objects). Holds 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65535 bytes of data
TEXT(size)	Holds a string with a maximum length of 65535 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16777215 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16777215 characters
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4294967295 bytes of data
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.)	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

Kanak Infosystems LLP.