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, or after a column in the table exists
ALL	Returns true if all of the subqueries values meet the condition
ALTER TABLE	Adds, deletes, or modifies columns in a table, or changes the data type of a column in a table
ALTER COLUMN	Changes the data type of a column in a table
AND	Logically joins two conditions in a WHERE clause, or in a JOIN statement
ANY	Returns true if any of the subqueries values meet the condition
AS	Renames a column or table with an alias
ASC	Sorts the result set in ascending order
BETWEEN	Selects values within a given range. The values can be numbers, text, or dates
CASE	Creates different outputs (like an IF-THEN-ELSE statement)
CHECK	Adds a check constraint on a column when a table is created, or on an existing column in an existing table
COLUMN	Changes, adds, or deletes a column in a table
CREATE DATABASE	Creates a database
CREATE INDEX	Creates an index in a table (allows you to create queries faster)
DELETE	Deletes rows in a table
DESC	Sorts the result set in descending order

SQL KEYWORDS

KEYWORDS	DESCRIPTION
DISTINCT	Selects only distinct values from a column
DROP	Deletes a column, a constraint, or a database
DROP DATABASE	Deletes a database
DROP INDEX	Deletes an index in a table
EXISTS	Tests for the existence of any record in a subquery
FROM	Specifies the table(s) from which data is selected or deleted
GROUP BY	Groups the result set by one or more columns
HAVING	Specifies a condition for a group or aggregate function to apply to
IN	Allows you to specify multiple values in a WHERE clause
INDEX	Creates and deletes indexes in a table
INNER JOIN	Returns rows that have matching values in both table
INSERT INTO	Inserts new rows in a table
IS NULL	Tests for empty values
IS NOT NULL	Tests for non-empty values
LIKE	Compares a value to similar values using wildcard operator
LIMIT	Specifies the number of records to return in the result set
NOT	Displays a record if the condition(s) is NOT true
OR	Logically joins two conditions in a WHERE clause, or in a JOIN statement
ORDER BY	Sorts the result set in ascending or descending order
OUTER JOIN	Returns rows that have matching values in both table
PRIMARY KEY	Creates a primary key on a table
PROCEDURE	Creates a stored procedure
RIGHT JOIN	Returns rows that have matching values in both table
ROWNUM	Specifies the number of records in the result set
SELECT	Selects data from a database
SELECT DISTINCT	Selects only distinct values from a column
SELECT TOP	Specifies the number of records to return in the result set
SET	Specifies which columns and values that should be updated in a table
SOME	Returns true if any of the subqueries values meet the condition
TOP	Specifies the number of records to return in the result set
TRUNCATE TABLE	Deletes the data inside a table, but not the table itself
UNION	Combines the result set of two or more SELECT statements (only distinct values)
UNION ALL	Combines the result set of two or more SELECT statements (allows duplicate values)
UNIQUE	Creates a unique constraint on a table when it is created, or on an existing table
UPDATE	Updates existing rows in a table
VALUES	Specifies the values or tuples to be inserted into a table
WHERE	Specifies a condition while fetching the data from a table, or joining multiple tables

Kanak Infosystems LLP

COMMENTS

SINGLE LINE COMMENTS

```
-- Select all:
SELECT * FROM city
```

MULTILINE COMMENTS

```
/* Select all
the records
in the table
-- */
SELECT * FROM city
```

MySQL Data Types

(version 8.0)

String types

Many types for many purposes.

STRING DATA TYPES

DATA TYPE	DESCRIPTION
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). Fixed size is specified in parenthesis. Can hold up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). Max size is specified in parenthesis in bytes. Can hold up to 65535 bytes
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(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 SET may contain up to 64 list items and can store more than one choice

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
BYTE(size)	Holds a small integer. Max size is specified in parenthesis in bytes. Signed range is from -128 to 127. Unsigned range is from 0 to 255
TINYINT(size)	Holds a very small integer. Max size is specified in parenthesis in bytes. Signed range is from -128 to 127. Unsigned range is from 0 to 255
SMALLINT(size)	Holds a small integer. Max size is specified in parenthesis in bytes. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535
MEDIUMINT(size)	Holds a medium integer. Max size is specified in parenthesis in bytes. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215
INT(size)	Holds a medium integer. Max size is specified in parenthesis in bytes. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295
BIGINT(size)	Holds a large integer. Max size is specified in parenthesis in bytes. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615
FLOAT(p)	Holds a floating point number. If p is between 0 and 24, it holds a 32 bit floating point number. If p is between 25 and 53, it holds a 64 bit floating point number
DOUBLE(size, d)	Holds a floating point number. Max number of digits are specified in size. Max number of digits after the decimal point are specified in d
DECIMAL(size, d), NUMERIC(size, d)	Holds a fixed point number. Max number of digits are specified in size. Max number of digits after the decimal point are specified in d

DATE AND TIME DATA TYPES

DATA TYPE	DESCRIPTION
DATE	Holds a date. Format: YYYY-MM-DD. Supported range is from '1000-01-01' to '9999-12-31'
DATETIME(fsp)	Holds a date and time. Format: YYYY-MM-DD hh:mm:ss. Supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
TIMESTAMP(fsp)	Holds a timestamp. Number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. Supported range is from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC
TIME(fsp)	Holds a time. Format: hh:mm:ss. Supported range is from '-838:59:59' to '838:59:59'
YEAR	Holds a year in a four-digit format. Allowed values: 1901 to 2155, and 0000

Kanak Infosystems LLP

OPERATORS

ARITHMETIC OPERATORS

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

COMPARISON OPERATORS

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

BITWISE OPERATORS

OPERATOR	DESCRIPTION
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
<<	Left shift
>>	Right shift
~	Bitwise not (1's complement)

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 a character x
CHAR_LENGTH(x)	Returns the length of a string x (in characters)
CHARACTER_LENGTH(x)	Returns the length of a 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 a value x in a list of values y,...
FIND_IN_SET(x,y)	Returns the position of a value x in a list of values y, in the form of a string
FORMAT(x,y)	Formats the numeric value x, to the format of 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 string y in string x
LCASE(x)	Converts a string x to lower-case
LEFT(x,y)	Returns the left y characters of string x
LENGTH(x)	Returns the length of a string x (in bytes)
LOCATE(x,y)	Returns the position of a value x in a string y
LOWER(x)	Converts a string x to lower-case
LPAD(x,y,z)	Adds a string z to the left of string x, for a total of y characters
LTRIM(x)	Removes leading spaces from a string x
MID(x,y,z)	Extracts a substring from a string (starting at position y, extending z characters long)
POSITION(x IN y)	Returns the position of a value x in a string y
REPEAT(x,y)	Repeats a string x, y times
REPLACE(x,y,z)	Replaces all occurrences of a string y in a string x, with a string z
REVERSE(x)	Reverses a string x, and returns the result
RIGHT(x,y)	Returns the right y characters of a string x
RPAD(x,y,z)	Adds a string z to the right of string x, for a total of y characters
RTRIM(x)	Removes trailing spaces from a string x
SPACE(x)	Returns a string x number of spaces
STRCMP(x,y)	Compares two strings x and y
SUBSTR(x,y,z), SUBSTRING(x,y,z)	Returns a substring z characters long, from a string x, starting at position y
SUBSTRING_INDEX(x,y,z)	Returns a substring of a string x, before z occurrences of a delimiter y
TRIM(x)	Removes leading and trailing spaces from a string x
UCASE(x)	Converts a string x to upper-case
UPPER(x)	Converts a string x to upper-case

NUMERIC FUNCTIONS

DATA TYPE	DESCRIPTION
ABS(x)	Returns the absolute value of a 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 an expression x
CEIL(x)	Rounds a number x up to the next whole number
CEILING(x)	Rounds a number x up to the next whole number
COS(x)	Returns the cosine of a number x
COT(x)	Returns the cotangent of a number x
COUNT(x)	Returns the number of records in a table
DIV(x,y)	Returns the integer quotient of a division of x by y
EXP(x)	Returns e raised to the power of x
FLOOR(x)	Rounds a number x down to the next whole number
GREATEST(x,y,...)	Returns the greatest value of the list of arguments
LN(x)	Returns the natural logarithm of a number x
LOG(x)	Returns the natural logarithm of a number x
LOG10(x)	Returns the base-10 logarithm of a number x
LOG2(x)	Returns the base-2 logarithm of a 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 of a number x 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 a number x to y decimal places
SIGN(x)	Returns the sign of a number x
SIN(x)	Returns the sine of a number x
SQRT(x)	Returns the square root of a number x
SUM(x)	Returns the total sum of an expression x
TAN(x)	Returns the tangent of a number x
TRUNCATE(x,y)	Returns x truncated to y decimal places

DATE FUNCTIONS

DATA TYPE	DESCRIPTION
ADDDATE(x,y)	Returns the date x and a date interval y added
ADDTIME(x,y)	Returns a time x and a time interval y 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()	Returns the date part from a date/time expression x
DATE_ADD(x,y)	Returns a date x and a date interval y added
DATE_FORMAT(x,y)	Formats a date x, according to the format specified in y
DATE_SUB(x,y)	Returns a date x and a date interval y subtracted
DAY(x)	Returns the day of the month from a date x
DAYNAME(x)	Returns the name of the weekday for a date x
DAYOFMONTH(x)	Returns the day of the month from a date x
DAYOFWEEK(x)	Returns the weekday index for a date x
DAYOFYEAR(x)	Returns the day of the year from a date x
FROM_DAYS(x)	Returns a date from a number x
HOUR(x)	Returns the hour from a date/time x
LAST_DAY(x)	Returns the last day of the month for a date x
LOCALTIME()	Returns the current date and time
LOCALTIMESTAMP()	Returns the current date and time
MAKEDATE(x,y)	Returns a date from a year x and a number of days y
MAKETIME(x,y,z)	Returns a time from an hour x, a minute y, and a second z
MICROSECOND(x)	Returns the microsecond part of a time x
MINUTE(x)	Returns the minute part of a time x
MONTH(x)	Returns the month part of a date x
MONTHNAME(x)	Returns the name of a month for a date x
NOW()	Returns the current date and time
PERIOD_ADD(x,y)	Returns a period x plus y number of months
PERIOD_DIFF(x,y)	Returns the difference between two periods x and y
QUARTER(x)	Returns the quarter for a date x
SECOND(x)	Returns the second part of a time x
SEC_TO_TIME(x)	Returns a time value from a given second x
STR_TO_DATE(x,y)	Returns a date from a string x, according to the format specified in y
SUBDATE(x,y)	Returns a date x, after a date interval y has been subtracted
SUBTIME(x,y)	Returns a time x, after a time interval y has been subtracted
SYSDATE()	Returns the current date and time
TIME()	Returns the time part from a given date/time x
TIME_FORMAT(x,y)	Formats a time x, according to the format specified in y
TIME_TO_SEC(x)	Returns the number of seconds from a 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 x and a time y
TO_DAYS(x)	Returns the number of days from a date x
WEEK(x)	Returns the week number for a given date
WEEKDAY(x)	Returns the weekday number for a given date
WEEKOFYEAR(x)	Returns the week number for a given date
YEAR(x)	Returns the year part of a given date
YEARWEEK(x)	Returns the year and week number for a given date

Kanak Infosystems LLP

KEYS

FOREIGN KEY

```
CREATE TABLE orders
(
    order_id int,
    order_date date,
    customer_id int,
    FOREIGN KEY (customer_id)
    REFERENCES customers(id)
);
```

```
CREATE TABLE products
(
    id int,
    name char(50),
    FOREIGN KEY (id)
    REFERENCES orders(order_id)
);
```

PRIMARY KEY

```
CREATE TABLE customers
(
    id int,
    last_name char(30),
    first_name char(30),
    address char(50),
    city char(30),
    country_id int,
    PRIMARY KEY (id)
);
```

Kanak Infosystems LLP

DATABASE

DATA TYPE	DESCRIPTION
CREATE DATABASE name	Creates a database with the specified name
DROP DATABASE name	Drops (deletes) a database and all its tables and views
CREATE DATABASE IF NOT EXISTS name	Creates a database with the specified name if it does not already exist
DROP DATABASE IF EXISTS name	Drops (deletes) a database and all its tables and views, if it exists
SHOW DATABASES	Displays a list of all databases on the server
USE name	Selects a database to work with

TABLES

DATA TYPE	DESCRIPTION
CREATE TABLE name (col1 type1, col2 type2, ...)	Creates a table with the specified name, and specified columns and data types
CREATE TABLE IF NOT EXISTS name (col1 type1, col2 type2, ...)	Creates a table with the specified name, and specified columns and data types, if it does not already exist
DROP TABLE name	Deletes a table, including all records, views, and privileges
DROP TABLE IF EXISTS name	Deletes a table, including all records, views, and privileges, if it exists
DESCRIBE name	Displays information about the columns in a table
SHOW TABLES	Displays a list of all tables in a database
SHOW COLUMNS FROM name	Displays information about the columns in a table
SHOW INDEX FROM name	Displays information about the indexes in a table
ALTER TABLE name ADD col type	Adds a column in a table
ALTER TABLE name DROP COLUMN col	Deletes a column in a table
ALTER TABLE name RENAME TO new_name	Changes the name of a table
ALTER TABLE name CHANGE col col type	Changes the data type of a column in a table
ALTER TABLE name MODIFY col type	Changes the data type of a column in a table
ALTER TABLE name ADD CONSTRAINT constraint	Adds a constraint after a table is created
ALTER TABLE name DROP CONSTRAINT constraint	Deletes a constraint
ALTER TABLE name ADD PRIMARY KEY (col)	Adds a primary key on a table when it is created
ALTER TABLE name DROP PRIMARY KEY	Deletes a primary key
ALTER TABLE name ADD FOREIGN KEY (col) REFERENCES table2(col2)	Adds a foreign key on a table when it is created
ALTER TABLE name DROP FOREIGN KEY fk	Deletes a foreign key
ALTER TABLE name ADD UNIQUE (col)	Adds a unique constraint on a table when it is created
ALTER TABLE name DROP INDEX index	Deletes a unique constraint
TRUNCATE TABLE name	Deletes the data inside a table, but not the table itself

VIEWS

DATA TYPE	DESCRIPTION
CREATE VIEW name AS SELECT col1, col2, ... FROM table WHERE condition;	Creates a view, based on a table, with the specified columns and records specified by the condition
CREATE VIEW name AS SELECT * FROM table WHERE condition;	Creates a view, based on a table, with all columns and records specified by the condition
CREATE OR REPLACE VIEW name AS SELECT col1, col2, ... FROM table WHERE condition;	Creates a view, based on a table, with the specified columns and records specified by the condition, or replaces an existing view
DROP VIEW name	Deletes a view
DROP VIEW IF EXISTS name	Deletes a view, if it exists

CREATING VIEWS

```
CREATE VIEW view_name AS
SELECT * FROM table_name;
```

REPLACING VIEWS

```
CREATE OR REPLACE VIEW view_name AS
SELECT * FROM table_name;
```

DELETING VIEWS

```
DROP VIEW view_name;
```

INDEXES

DATA TYPE	DESCRIPTION
CREATE INDEX name ON table (col1, col2, ...)	Creates an index, based on the specified columns
CREATE UNIQUE INDEX name ON table (col1, col2, ...)	Creates a unique index, based on the specified columns
DROP INDEX name ON table	Deletes an index
SHOW INDEX FROM table	Displays information about the indexes in a table

JOINS

ORDERS

id	date	customer_id	total_amount
1	2020-01-01	1	100
2	2020-01-02	2	200
3	2020-01-03	3	300

PRODUCTS

id	name	manufacturer	price
1	Product 1	Company A	100
2	Product 2	Company B	200
3	Product 3	Company C	300

CROSS

id	name	manufacturer	price
1	Product 1	Company A	100
1	Product 1	Company B	200
1	Product 1	Company C	300
2	Product 2	Company A	100
2	Product 2	Company B	200
2	Product 2	Company C	300
3	Product 3	Company A	100
3	Product 3	Company B	200
3	Product 3	Company C	300

SELF JOIN

```
SELECT a.name, b.name
FROM table a, table b
WHERE a.id = b.id
```

ALIAS

```
SELECT * FROM table1 AS t1
INNER JOIN table2 AS t2
ON t1.id = t2.id;
```

UNION

```
SELECT * FROM table1
UNION
SELECT * FROM table2;
```

UNION ALL

```
SELECT * FROM table1
UNION ALL
SELECT * FROM table2;
```

INTERSECT

```
SELECT * FROM table1
INTERSECT
SELECT * FROM table2;
```

MINUS

```
SELECT * FROM table1
MINUS
SELECT * FROM table2;
```

Kanak Infosystems LLP

INDEXES

DATA TYPE	DESCRIPTION
CREATE INDEX name ON table (col1, col2, ...)	Creates an index, based on the specified columns
CREATE UNIQUE INDEX name ON table (col1, col2, ...)	Creates a unique index, based on the specified columns
DROP INDEX name ON table	Deletes an index
SHOW INDEX FROM table	Displays information about the indexes in a table

ALTER TABLE

DATA TYPE	DESCRIPTION
ALTER TABLE name ADD col type	Adds a column in a table
ALTER TABLE name DROP COLUMN col	Deletes a column in a table
ALTER TABLE name RENAME TO new_name	Changes the name of a table
ALTER TABLE name CHANGE col col type	Changes the data type of a column in a table
ALTER TABLE name MODIFY col type	Changes the data type of a column in a table
ALTER TABLE name ADD CONSTRAINT constraint	Adds a constraint after a table is created
ALTER TABLE name DROP CONSTRAINT constraint	Deletes a constraint
ALTER TABLE name ADD PRIMARY KEY (col)	Adds a primary key on a table when it is created
ALTER TABLE name DROP PRIMARY KEY	Deletes a primary key
ALTER TABLE name ADD FOREIGN KEY (col) REFERENCES table2(col2)	Adds a foreign key on a table when it is created
ALTER TABLE name DROP FOREIGN KEY fk	Deletes a foreign key
ALTER TABLE name ADD UNIQUE (col)	Adds a unique constraint on a table when it is created
ALTER TABLE name DROP INDEX index	Deletes a unique constraint
TRUNCATE TABLE name	Deletes the data inside a table, but not the table itself

CONSTRAINTS

DATA TYPE	DESCRIPTION
CREATE TABLE name (col1 type1 constraint, col2 type2 constraint, ...)	Creates a table with the specified name, and specified columns and data types, with the specified constraints
CREATE TABLE IF NOT EXISTS name (col1 type1 constraint, col2 type2 constraint, ...)	Creates a table with the specified name, and specified columns and data types, with the specified constraints, if it does not already exist
NOT NULL	Indicates that a column cannot store NULL values
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	Uniquely identifies each record in a table. Must be unique and cannot be NULL
FOREIGN KEY	Prevents actions that would destroy links between tables
CHECK	Ensures that all values in a column satisfy a specified condition
DEFAULT	Sets a default value for a column when no value is specified
AUTO_INCREMENT	Generates a unique number when a new record is inserted into a table
CONSTRAINT	Creates a constraint after a table is created
ADD CONSTRAINT	Adds a constraint after a table is created
DROP CONSTRAINT	Deletes a constraint

VIEWS

DATA TYPE	DESCRIPTION
CREATE VIEW name AS SELECT col1, col2, ... FROM table WHERE condition;	Creates a view, based on a table, with the specified columns and records specified by the condition
CREATE VIEW name AS SELECT * FROM table WHERE condition;	Creates a view, based on a table, with all columns and records specified by the condition
CREATE OR REPLACE VIEW name AS SELECT col1, col2, ... FROM table WHERE condition;	Creates a view, based on a table, with the specified columns and records specified by the condition, or replaces an existing view
DROP VIEW name	Deletes a view
DROP VIEW IF EXISTS name	Deletes a view, if it exists

CREATING VIEWS

```
CREATE VIEW view_name AS
SELECT * FROM table_name;
```

REPLACING VIEWS

```
CREATE OR REPLACE VIEW view_name AS
SELECT * FROM table_name;
```

DELETING VIEWS

```
DROP VIEW view_name;
```

JOINS

ORDERS

id	date	customer_id	total_amount
1	2020-01-01	1	100
2	2020-01-02	2	200
3	2020-01-03	3	300

PRODUCTS

id	name	manufacturer	price
1	Product 1	Company A	100
2	Product 2	Company B	200
3	Product 3	Company C	300

CROSS

id	name	manufacturer	price
1	Product 1	Company A	100
1	Product 1	Company B	200
1	Product 1	Company C	300
2	Product 2	Company A	100
2	Product 2	Company B	200
2	Product 2	Company C	300
3	Product 3	Company A	100
3	Product 3	Company B	200
3	Product 3	Company C	300

SELF JOIN

```
SELECT a.name, b.name
FROM table a, table b
WHERE a.id = b.id
```

ALIAS

```
SELECT * FROM table1 AS t1
INNER JOIN table2 AS t2
ON t1.id = t2.id;
```

UNION

```
SELECT * FROM table1
UNION
SELECT * FROM table2;
```

UNION ALL

```
SELECT * FROM table1
UNION ALL
SELECT * FROM table2;
```

INTERSECT

```
SELECT * FROM table1
INTERSECT
SELECT * FROM table2;
```

MINUS

```
SELECT * FROM table1
MINUS
SELECT * FROM table2;
```

Kanak Infosystems LLP

www.kanakinfosystems.com