Kanak

SQL

```
SELECT * FROM PERSONS;
SELECT * FROM PERSONS
WHERE ID_P = 2
ORDER BY LastName;
SELECT * FROM PERSONS
WHERE ID_P = 2
AND FirstName='Tove';
```

SQL KEYWORDS

KEYWORDS	DESCRIPTION
ADD	Adding a column in an existing table, Adding a constraint after a table is created, Adding a value to an enum type
ADD CONSTRAINT	Adding a constraint after a table is created, Adding a primary key after a table is created
ALTER TABLE	Adding a column in an existing table, Adding a constraint after a table is created, Adding a primary key after a table is created, Changing the data type of a column in a table, Renaming a column in a table, Renaming a table
ALTER COLUMN	Changing the data type of a column in a table, Renaming a column in a table
ALL	Comparing a value to all values in another field, Returning all fields in a table, Returning all values in a field
AND	Combining two conditions, Displaying records in a table, Joining tables
ANY	Comparing a value to any value in another field, Returning any value in a field
AS	Assigning an alias to a table, Assigning an alias to a column, Creating a view of a table
ASC	Displaying records in ascending order
BETWEEN	Comparing a value to a range of values, Displaying records in a table, Joining tables
CASE	Creating a conditional statement, Displaying records in a table, Joining tables
CHECK	Adding a constraint after a table is created, Adding a constraint when a table is created
CREATE DATABASE	Creating a database, Displaying a database
CREATE INDEX	Creating an index, Displaying an index
DEFAULT	Adding a default value to a column in an existing table, Adding a default value when a table is created
DELETE	Deleting records from a table, Joining tables
DESC	Displaying records in descending order

SQL KEYWORDS

KEYWORDS	DESCRIPTION
DISTINCT	Displaying unique values in a field, Displaying unique records in a table, Joining tables
DROP	Deleting a column in an existing table, Deleting a constraint after a table is created, Deleting a database, Deleting an index, Deleting a table, Deleting a view
DROP CONSTRAINT	Deleting a constraint after a table is created
DROP DATABASE	Deleting a database
DROP INDEX	Deleting an index
DROP TABLE	Deleting a table
DROP VIEW	Deleting a view
EXISTS	Checking for existence of a record in a table, Joining tables
FROM	Selecting a table to select records from, Joining tables
GROUP BY	Grouping records in a table, Joining tables
HAVING	Selecting groups of records in a table, Joining tables
IN	Comparing a value to a list of values, Joining tables
INDEX	Creating an index, Displaying an index
INNER JOIN	Joining tables
INSERT INTO	Inserting records into a table, Joining tables
IS NULL	Testing for a null value, Joining tables
IS NOT NULL	Testing for a non-null value, Joining tables
JOIN	Joining tables
LEFT JOIN	Joining tables
LIKE	Comparing a value to similar values, Joining tables
LIMIT	Specifying the maximum number of records to return, Joining tables
NOT	Combining two conditions, Displaying records in a table, Joining tables
NOT NULL	Adding a constraint after a table is created, Adding a constraint when a table is created
OR	Combining two conditions, Displaying records in a table, Joining tables
ORDER BY	Sorting records in a table, Joining tables
OUTER JOIN	Joining tables
PRIMARY KEY	Adding a primary key after a table is created, Adding a primary key when a table is created
RIGHT JOIN	Joining tables
ROWNUM	Selecting a specified number of records in a table, Joining tables
SELECT	Selecting records from a table, Joining tables
SELECT DISTINCT	Selecting unique records from a table, Joining tables
SELECT TOP	Selecting a specified number of records from a table, Joining tables
SET	Specifying new values for records in a table, Joining tables
SOME	Comparing a value to some values in another field, Returning some values in a field
TOP	Selecting a specified number of records from a table, Joining tables
TRUNCATE TABLE	Deleting records from a table, Joining tables
UNION	Combining the results of two or more queries, Joining tables
UNION ALL	Combining the results of two or more queries, Joining tables
UNIQUE	Adding a constraint after a table is created, Adding a constraint when a table is created
UPDATE	Updating records in a table, Joining tables
VALUES	Specifying the values for an INSERT INTO statement, Joining tables
WHERE	Filtering records in a table, Joining tables

Kanak Infosystems LLP

COMMENTS

SINGLE LINE COMMENTS

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

MULTILINE COMMENTS

```
/* Select all
from city table
and order them by name */
SELECT * FROM city
ORDER BY name;
```

MySQL Data Types

(version 8.0)

NUMERIC DATA TYPES

DATA TYPE	DESCRIPTION
TINYINT(M)	Signed: -128 to 127. Unsigned: 0 to 255. Optional parameter M specifies the number of digits to display
SMALLINT(M)	Signed: -32768 to 32767. Unsigned: 0 to 65535. Optional parameter M specifies the number of digits to display
MEDIUMINT(M)	Signed: -8388608 to 8388607. Unsigned: 0 to 16777215. Optional parameter M specifies the number of digits to display
INT(M)	Signed: -2147483648 to 2147483647. Unsigned: 0 to 4294967295. Optional parameter M specifies the number of digits to display
BIGINT(M)	Signed: -9223372036854775808 to 9223372036854775807. Unsigned: 0 to 18446744073709551615. Optional parameter M specifies the number of digits to display
DECIMAL(M,D)	Fixed-point number. M specifies the total number of digits. D specifies the number of digits after the decimal point
FLOAT(p)	Floating point number. p specifies the precision (0-24 for 4 bytes, 25-53 for 8 bytes)
DOUBLE(M,D), DOUBLE PRECISION(M,D)	Floating point number. M specifies the total number of digits. D specifies the number of digits after the decimal point
BIT(M)	Bit-value type. M specifies the number of bits per value (from 1 to 64)
BOOL, BOOLEAN	Synonyms for TINYINT(1)

NOTABLE DATA TYPES

DATA TYPE	DESCRIPTION
BYTEA(n)	Variable-length binary string. Optional parameter n specifies the maximum length in bytes
TINYTEXT(M), TEXT(M), MEDIUMTEXT(M), LONGTEXT(M)	Variable-length string. Optional parameter M specifies the maximum length in characters
ENUM('value1','value2',...)	Enumeration. Can store a maximum of 65535 distinct values. Internally represented as integers
SET('value1','value2',...)	Set. Can store any combination of up to 64 distinct values. Internally represented as integers
BINARY(n), VARBINARY(n)	Binary byte string. Optional parameter n specifies the maximum length in bytes
TINYBLOB(M), BLOB(M), MEDIUMBLOB(M), LONGBLOB(M)	Binary large object. Optional parameter M specifies the maximum length in characters
JSON	Stores JSON (JavaScript Object Notation) data
GEOMETRY(shape,srid)	Stores geometry data. Optional parameter shape specifies the shape, and srid specifies the SRID (Spatial Reference Identifier)

Kanak Infosystems LLP

OPERATORS

ARITHMETIC OPERATORS

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

BITWISE OPERATORS

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

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 OR equal
^=	Bitwise XOR equal
*=	Bitwise shift left equal
>>=	Bitwise shift right equal
~=	Bitwise NOT 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
CHARACTER_LENGTH(x)	Returns the length of the string x
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 first occurrence of the string x in the string list 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 occurrence of string y in string x
LCASE(x)	Converts a string x to lower-case
LEFT(x,y)	Returns the left y characters of the string x
LENGTH(x)	Returns the length of the string x in bytes
LOCATE(x,y)	Returns the first occurrence of the string x in string y
LOWER(x)	Converts a string x to lower-case
LPAD(x,y,z)	Left pads the string x by the string z, for a length of y characters
LTRIM(x)	Removes leading spaces from the string x
MID(x,y,z)	Returns a substring starting from position y, with length z, from the string x
POSITION(x IN y)	Returns 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 and returns the result
RIGHT(x,y)	Returns the right y characters of the string x
RPAD(x,y,z)	Right pads the string x by the string z, for a length of y characters
RTRIM(x)	Removes trailing spaces from the string x
SPACE(x)	Returns x number of spaces
STRCMP(x,y)	Compares two strings x and y
SUBSTR(x,y,z), SUBSTRING(x,y,z)	Returns a substring of the string x, starting from position y, with length z
SUBSTRING_INDEX(x,y,z)	Returns a substring of the string x before z occurrences of the delimiter y
TRIM(x)	Removes leading and trailing spaces from the 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 the given number x
ACOS(x)	Returns the arc cosine of the given number x
ASIN(x)	Returns the arc sine of the given number x
ATAN(x)	Returns the arc tangent of the given number x
ATAN2(x,y)	Returns the arc tangent of the given parameters 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
DIV(x,y)	Returns x DIV y
EXP(x)	Returns e to the x-th power
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 given field
MIN(x)	Returns the minimum value in a given field
MOD(x,y)	Returns x MOD y
PI()	Returns PI
POW(x,y)	Returns the value of x raised to the power of y
POWER(x,y)	Returns the value of 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 the given number x
SQRT(x)	Returns the square root of the given number x
SUM(x)	Returns the total sum of a given field
TAN(x)	Returns the tangent of the given number x
TRUNCATE(x,y)	Returns x truncated to y decimal places

DATE FUNCTIONS

DATA TYPE	DESCRIPTION
ADDDATE(x,y)	Returns the date x with the time or days interval y added
ADDTIME(x,y)	Returns the time x with the 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(x)	Returns the date part of the date x
DATE_FORMAT(x,y)	Formats the date x according to the format specified in y
DATEDIFF(x,y)	Returns the difference between the date x and the date y
DAY(x)	Returns the day of the month for 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 x days from the start of year 0
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
MAKEDATE(x,y)	Returns a date from the year x and a day number y
MAKETIME(x,y,z)	Returns a time from the hour x, minute y, and second z
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 a period x plus 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
SEC_TO_TIME(x)	Returns the time for the given seconds x
STR_TO_DATE(x,y)	Returns a date based on the given string x and format y
SUBDATE(x,y)	Returns the date x with the time or days interval y subtracted
SUBTIME(x,y)	Returns the time x with the time interval y subtracted
SYSDATE()	Returns the current date and time
TIME(x)	Returns the time part of the given date x
TIME_FORMAT(x,y)	Formats the time x according to the format specified in y
TIME_TO_SEC(x)	Returns the seconds for 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 based on the date or datetime x and the time y
TO_DAYS(x)	Returns the number of days for the given date x
WEEK(x)	Returns the week number for the given date x
WEEKDAY(x)	Returns the weekday index 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

```
CREATE TABLE PRODUCTS
(P_Id int NOT NULL,
ProductName varchar(15) NOT NULL,
UnitPrice int NULL,
UnitInStock int NULL,
PRIMARY KEY (P_Id)
)
```

```
CREATE TABLE ORDERS
(O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
PRIMARY KEY (O_Id),
FOREIGN KEY (P_Id) REFERENCES persons(P_Id)
)
```

PRIMARY KEY

```
CREATE TABLE Persons
(P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
PRIMARY KEY (P_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
DROP DATABASE [IF EXISTS] name	Drops (deletes) a database and all its tables and views. Optional IF EXISTS statement prevents errors from occurring if the database does not exist
SHOW DATABASES	Lists the databases that are present on the server
USE name	Switches to a database

TABLES

DATA TYPE	DESCRIPTION
CREATE TABLE name (col1 type1, col2 type2, ...)	Creates a table with the specified name and the specified columns and data types
CREATE TABLE IF NOT EXISTS name (col1 type1, col2 type2, ...)	Creates a table with the specified name and the specified columns and data types, if it does not already exist
DROP TABLE name;	Deletes a table
DROP TABLE IF EXISTS name;	Deletes a table, if it exists
SHOW TABLES;	Lists the tables in the current database
SHOW TABLES LIKE 'name';	Lists the tables in the current database that match the specified name
DESCRIBE name;	Lists the columns in a table, and their data types and keys
INSERT INTO name VALUES (val1, val2, ...);	Inserts a record into a table
INSERT INTO name (col1, col2, ...) VALUES (val1, val2, ...);	Inserts a record into a table, for the specified columns
INSERT INTO name VALUES (val1, val2, ...), (val1, val2, ...), ...;	Inserts multiple records into a table
INSERT INTO name (col1, col2, ...) VALUES (val1, val2, ...), (val1, val2, ...), ...;	Inserts multiple records into a table, for the specified columns
INSERT INTO name1 SELECT * FROM name2;	Inserts records into a table from another table
INSERT INTO name1 (col1, col2, ...) SELECT col1, col2, ... FROM name2;	Inserts records into a table from another table, for the specified columns
SELECT * FROM name;	Selects records from a table
SELECT col1, col2, ... FROM name;	Selects records from a table, for the specified columns
SELECT * FROM name1 JOIN name2 ON name1.col1 = name2.col1;	Selects records from two tables, where the specified columns are equal
SELECT * FROM name1 LEFT JOIN name2 ON name1.col1 = name2.col1;	Selects records from two tables, where the specified columns are equal, and all records from the first table
SELECT * FROM name1 RIGHT JOIN name2 ON name1.col1 = name2.col1;	Selects records from two tables, where the specified columns are equal, and all records from the second table
SELECT * FROM name1 FULL JOIN name2 ON name1.col1 = name2.col1;	Selects records from two tables, where the specified columns are equal, and all records from both tables
SELECT * FROM name1 CROSS JOIN name2;	Selects records from two tables, and returns all possible combinations
SELECT * FROM name1, name2;	Selects records from two tables, and returns all possible combinations
SELECT * FROM name1 NATURAL JOIN name2;	Selects records from two tables, where the specified columns are equal
SELECT * FROM name1 WHERE col1 IN (SELECT col1 FROM name2);	Selects records from a table, where the specified records are found in another table
SELECT * FROM name1 WHERE EXISTS (SELECT * FROM name2);	Selects records from a table, if a specified table has any records
SELECT * FROM name1 WHERE col1 > ANY (SELECT col1 FROM name2);	Selects records from a table, where the specified records are greater than any of the records in another table
SELECT * FROM name1 WHERE col1 > ALL (SELECT col1 FROM name2);	Selects records from a table, where the specified records are greater than all of the records in another table
SELECT * FROM name1 WHERE col1 = (SELECT col1 FROM name2);	Selects records from a table, where the specified records are equal to the records in another table
SELECT * FROM name1 WHERE col1 LIKE 'pattern';	Selects records from a table, where the specified records match the specified pattern
SELECT * FROM name1 WHERE col1 IN ('val1', 'val2', ...);	Selects records from a table, where the specified records are found in the specified values
SELECT * FROM name1 WHERE col1 BETWEEN 'val1' AND 'val2';	Selects records from a table, where the specified records are between the specified values
SELECT * FROM name1 WHERE col1 IS NULL;	Selects records from a table, where the specified records are null
SELECT * FROM name1 WHERE col1 IS NOT NULL;	Selects records from a table, where the specified records are not null
SELECT * FROM name1 ORDER BY col1 ASC;	Selects records from a table, and sorts the records by the specified columns, in ascending order
SELECT * FROM name1 ORDER BY col1 DESC;	Selects records from a table, and sorts the records by the specified columns, in descending order
SELECT * FROM name1 ORDER BY col1 ASC, col2 DESC;	Selects records from a table, and sorts the records by the specified columns, in ascending order and descending order
SELECT * FROM name1 ORDER BY col1 ASC LIMIT 5;	Selects records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records
SELECT * FROM name1 ORDER BY col1 ASC LIMIT 5 OFFSET 2;	Selects records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT * FROM name1 ORDER BY col1 ASC LIMIT 2, 5;	Selects records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT * FROM name1 ORDER BY col1 ASC LIMIT 5 ROWS FETCH NEXT 1 ROW ONLY;	Selects records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT * FROM name1 ORDER BY col1 ASC LIMIT 5 ROWS FETCH NEXT 1 ROW WITH TIES;	Selects records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record, and also records that are equal to the last record
SELECT DISTINCT col1, col2, ... FROM name;	Selects unique records from a table, for the specified columns
SELECT TOP 5 * FROM name;	Selects the first 5 records from a table
SELECT TOP 5 * FROM name1 JOIN name2 ON name1.col1 = name2.col1;	Selects the first 5 records from two tables, where the specified columns are equal
SELECT TOP 5 * FROM name1 LEFT JOIN name2 ON name1.col1 = name2.col1;	Selects the first 5 records from two tables, where the specified columns are equal, and all records from the first table
SELECT TOP 5 * FROM name1 RIGHT JOIN name2 ON name1.col1 = name2.col1;	Selects the first 5 records from two tables, where the specified columns are equal, and all records from the second table
SELECT TOP 5 * FROM name1 FULL JOIN name2 ON name1.col1 = name2.col1;	Selects the first 5 records from two tables, where the specified columns are equal, and all records from both tables
SELECT TOP 5 * FROM name1 CROSS JOIN name2;	Selects the first 5 records from two tables, and returns all possible combinations
SELECT TOP 5 * FROM name1, name2;	Selects the first 5 records from two tables, and returns all possible combinations
SELECT TOP 5 * FROM name1 NATURAL JOIN name2;	Selects the first 5 records from two tables, where the specified columns are equal
SELECT TOP 5 * FROM name1 WHERE col1 IN (SELECT col1 FROM name2);	Selects the first 5 records from a table, where the specified records are found in another table
SELECT TOP 5 * FROM name1 WHERE EXISTS (SELECT * FROM name2);	Selects the first 5 records from a table, if a specified table has any records
SELECT TOP 5 * FROM name1 WHERE col1 > ANY (SELECT col1 FROM name2);	Selects the first 5 records from a table, where the specified records are greater than any of the records in another table
SELECT TOP 5 * FROM name1 WHERE col1 > ALL (SELECT col1 FROM name2);	Selects the first 5 records from a table, where the specified records are greater than all of the records in another table
SELECT TOP 5 * FROM name1 WHERE col1 = (SELECT col1 FROM name2);	Selects the first 5 records from a table, where the specified records are equal to the records in another table
SELECT TOP 5 * FROM name1 WHERE col1 LIKE 'pattern';	Selects the first 5 records from a table, where the specified records match the specified pattern
SELECT TOP 5 * FROM name1 WHERE col1 IN ('val1', 'val2', ...);	Selects the first 5 records from a table, where the specified records are found in the specified values
SELECT TOP 5 * FROM name1 WHERE col1 BETWEEN 'val1' AND 'val2';	Selects the first 5 records from a table, where the specified records are between the specified values
SELECT TOP 5 * FROM name1 WHERE col1 IS NULL;	Selects the first 5 records from a table, where the specified records are null
SELECT TOP 5 * FROM name1 WHERE col1 IS NOT NULL;	Selects the first 5 records from a table, where the specified records are not null
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order
SELECT TOP 5 * FROM name1 ORDER BY col1 DESC;	Selects the first 5 records from a table, and sorts the records by the specified columns, in descending order
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC, col2 DESC;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order and descending order
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC LIMIT 5;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC LIMIT 5 OFFSET 2;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC LIMIT 2, 5;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC LIMIT 5 ROWS FETCH NEXT 1 ROW ONLY;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record
SELECT TOP 5 * FROM name1 ORDER BY col1 ASC LIMIT 5 ROWS FETCH NEXT 1 ROW WITH TIES;	Selects the first 5 records from a table, and sorts the records by the specified columns, in ascending order, and returns the first 5 records, starting from the 2 record, and also records that are equal to the last record
SELECT DISTINCT col1, col2, ... FROM name;	Selects unique records from a table, for the specified columns