

## 1. Constraints:

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK
6. UNIQUE
7. DEFAULT
8. CREATE INDEX

## 2. Data Types:

### Numeric:

- |               |                             |
|---------------|-----------------------------|
| 1. BIGINT     | $-2^{63}$ to $2^{63}-1$     |
| 2. INT        | $-2^{31}$ to $2^{31}-1$     |
| 3. SMALLINT   | $-2^{15}$ to $2^{15}-1$     |
| 4. TINYINT    | 0 to $2^8-1$                |
| 5. BIT        | 0 to 1                      |
| 6. DECIMAL    | $-10^{38}+1$ to $10^{38}-1$ |
| 7. NUMERIC    | $-10^{38}+1$ to $10^{38}-1$ |
| 8. MONEY      |                             |
| 9. SMALLMONEY |                             |
| 10. FLOAT     |                             |
| 11. DOUBLE    |                             |
| 12. NUMBER    |                             |

### String:

- |                 |                                       |
|-----------------|---------------------------------------|
| 1. CHAR         | (upto 8000 chars, fixed length)       |
| 2. VARCHAR      | (upto 8000 chars, variable length)    |
| 3. VARCHAR(MAX) | (upto 231 chars, only in 2005 server) |
| 4. TEXT         | (upto 2,127,483,647)                  |

### Date and Time:

- |         |            |
|---------|------------|
| 1. DATE | YYYY-MM-DD |
|---------|------------|

- |              |                     |
|--------------|---------------------|
| 2. DATETIME  | YYYY-MM-DD hh:mm:ss |
| 3. TIMESTAMP |                     |
| 4. TIME      | hh:mm:ss            |
| 5. YEAR      | YYYY                |

### **3. Aggregate Functions:**

Returns Single value by performing some calculation on set of values

Used with GROUP BY clause OR SELECT statement

1. MIN()
2. MAX()
3. SUM()
4. AVG()
5. COUNT()

### **4. Commands:**

#### **DDL**

1. DROP

Syntax: DROP TABLE tableName;

2. RENAME

Syntax: RENAME TABLE oldTableName to newTablename;

ALTER TABLE oldTableName RENAME newTableName;

3. CREATE

Syntax: CREATE TABLE tableName (colName dType, colName dType, colname dType);

4. ALTER

- a. ADD COLUMN

Syntax: ALTER TABLE tableName ADD COLUMN columnName dType;

- b. DROP COLUMN

Syntax: ALTER TABLE tableName DROP COLUMN columnName;

- c. RENAME COLUMN

Syntax: ALTER TABLE tableName RENAME COLUMN oldColName to newColName;

- d. MODIFY COLUMN

Syntax: ALTER TABLE tableName MODIFY COLUMN colName dType;

## 5. TRUNCATE

Syntax: TRUNCATE TABLE tableName;

## 6. COMMENT

### **DQL**

## 1. SELECT

### **DML**

## 1. INSERT

Syntax: INSERT INTO tableName (colName, colName,...) VALUES (val1, val2,...);

## 2. UPDATE

Syntax: UPDATE tableName set col1, col2,.. = val WHERE condition;

## 3. DELETE

Syntax: DELETE FROM tableName WHERE condition;

### **DCL**

## 1. GRANT

## 2. REVOKE

### **TCL**

## 1. BEGIN TRANSACTION

## 2. COMMIT

## 3. ROLLBACK

## 4. SAVEPOINT

## **5. Important Keywords:**

## 1. SELECT

## 2. SELECT DISTINCT

## 3. WHERE

## 4. ORDER BY

## 5. AND

## 6. OR

## 7. NOT

## 8. INSERT INTO

## 9. IS NULL

10. IS NOT NULL
11. UPDATE
12. DELETE
13. SELECT TOP (ROWNUM)
14. LIMIT
15. AGGREGATE FUNCTIONS (MIN, MAX, SUM, AVG, COUNT)
16. LIKE
17. WILDCARDS
18. IN
19. BETWEEN
20. AS
21. JOIN , LEFT JOIN, RIGHT JOIN, SELF JOIN, FULL OUTER JOIN (NULL values will be there)
22. UNION
23. GROUP BY
24. HAVING
25. EXISTS
26. ANY (SOME), ALL
27. CASE WHEN cond THEN someValue ELSE someValue END
28. INSERT INTO SELECT
29. IFNULL(), ISNULL(), COALESCE(), NVL()

All these are used for same, when we may encounter the null values

Old query: **SELECT a\*b from table1;**

Here a is definitely a number, but b contains some null values also then the product becomes NULL.

So to avoid that one we will use these Null Functions.

New query: **SELECT a \* IFNULL(b, 0) from table1;** (MY SQL)

New query: **SELECT a \* ISNULL(b, 0) from table1;** (SQL SERVER)

New query: **SELECT a \* COALESCE(b, 0) from table1;** (\*)

New query: **SELECT a \* NVL(b, 0) from table1;** (ORACLE)

So COALESCE applicable to all

30. Stored Procedures

Syntax: CREATE PROCEDURE procedureName AS selectStatement:

31. CONCAT
32. GROUP\_CONCAT

33. ADD

34. ADD CONSTRAINT

35. ALTER

36. ALTER COLUMN

37. ALTER TABLE

38. ASC

39. BACKUP DATABASE

40. CONSTRAINT

41. DESC

42. DISTINCT

43. DROP CONSTRAINT

44. DROP INDEX

45. DROP VIEW

46. EXEC (Executes stored procedure)

47. EXISTS

48. FOREIGN KEY

49. INDEX

50. RANK() (Same rank for same values)

Syntax: SELECT RANK() OVER(ORDER BY colName) AS aliasname FROM tableName

Ex: 90---1

90---1

80---2

51. ROW\_NUMBER() (Unique and Continuous)

Learn Row\_Number with partition by \*\*\*

Syntax: SELECT RANK() OVER(ORDER BY colName) AS aliasname FROM tableName

Ex: 90---1

90---2

80---3

52. DENSE\_RANK() (Not Unique, means assign same rank for same value and Not continuous)

Ex: 90---1

90---1

80---3

53. NTILE() (We can give how many ranks we require, then it will split)

Syntax: SELECT NTILE(ranksRequired) OVER(ORDER BY colName) AS aliasname FROM tableName

## 6. Wild Cards:

Like regex. For substituting the characters.

% → More than one character

\_ → Single character

## 7. SQL Mode:

SET sql\_mode = 'STRICT\_TRANS\_TABLES'

For avoiding conflicts while using the group by clause.

---

# SQL Server Functions

## 1. String Functions:

### 1. ASCII

Ex: SELECT ASCII(colName) from tableName;

### 2. CHAR

Ex: SELECT CHAR(65) AS codeToCharacter;                      A

### 3. CHARINDEX

Ex: SELECT CHARINDEX('t', 'CUSTOMER') matchPos;            0

### 4. CONCAT

Ex: SELECT CONCAT('W3SCHOOLS', '.COM');

### 5. CONCAT WITH +

Ex: SELECT 'W3SCHOOLS'+'COM';

### 6. CONCAT\_WS (concat with separator)

Ex: SELECT CONCAT\_WS('.', 'WWW', 'W3SCHOOLS', 'COM');

### 7. DATALENGTH

Ex: SELECT DATALENGTH('JUICE');                              5

### 8. DIFFERENCE

Ex: SELECT DIFFERENCE('JUICE', 'JUCY');

Return from 0(no similarity) to 4(high similarity)

### 9. FORMAT

### 10. LEFT

Ex: SELECT LEFT('HEMCHAND', 3);

#### 11. LEN

Ex: SELECT LEN('HEMCHAND');

#### 12. LOWER

Ex: SELECT LOWER('HEMCHAND');

#### 13. LTRIM

Ex: SELECT LTRIM(' HEMCHAND');

#### 14. NCHAR

Ex: SELECT NCHAR(65);

#### 15. PATINDEX(patternIndex)

Ex: SELECT PATINDEX('%AND%', 'HEMCHAND');

#### 16. REPLACE

Ex: SELECT REPLACE('SQL TUTORIAL', 'T', 'M');

SQL MUMORIAL

#### 17. REVERSE

Ex: SELECT REVERSE('HEMCHAND');

#### 18. RIGHT

Ex: SELECT RIGHT('HEMCHAND', 3);

#### 19. RTRIM

Ex: SELECT RTRIM('HEMCHAND ');

#### 20. SPACE

Ex: SELECT SPACE(10)

' '

#### 21. STR

Ex: SELECT STR(185)

'185'

#### 22. STUFF

Ex: SELECT STUFF('SQL TUTORIAL', 1,3, 'HTML');

'HTML TUTORIAL'

#### 23. SUBSTRING

Ex: SELECT SUBSTRING('SQL TUTORIAL', 1, 3);

'SQL'

#### 24. TRIM

Ex: SELECT TRIM(' HEMCHAND ')

'HEMCHAND'

#### 25. UNICODE (returns Unicode value of first char)

Ex: SELECT UNICODE ('ATLANTA');

65

#### 26. UPPER

Ex: SELECT UPPER('hemchand');

'HEMCHAND'

## 2. Numeric Functions:

1. ABS

Ex: SELECT ABS(-243.5);

2. ACOS

3. ASIN

4. ATAN

5. ATN2

Ex: SELECT ATN2(0.50, 1);

6. AVG

7. CEILING

Ex: SELECT CEILING(25.75);

8. COUNT

9. COS

10. COT

11. DEGREES

Ex: SELECT DEGREES(1.5);

12. EXP

Ex: SELECT EXP(1);

13. FLOOR

Ex: SELECT FLOOR(1.7);

14. LOG

15. LOG10

16. MAX

17. MIN

18. PI

Ex: SELECT PI();

19. POWER

20. RADIANS

Ex: SELECT RADIAN(180);

21. RAND

Ex: SELECT RAND();



## 22. ROUND

Ex: `SELECT ROUND(243.415, 2);`

23. SIGN (returns sign of number)

## 24. SIN

## 25. SQRT

## 26. SQUARE

## 27. SUM

## 28. TAN

### 3. Date Functions:

1. CURRENT\_TIMESTAMP

Ex: SELECT CURRENT\_TIMESTAMP;                      YYYY-MM-DD hh:mm:ss.

## 2. DATEADD

Ex: `SELECT DATEADD(year, 1, '2017/08/25')` `'2018/08/25 00:00:00.000'`

Ex: `SELECT DATEADD(month, 20, '2017/08/25')`      `'2019/04/25 00:00:00.000'`

### 3. DATEDIFF

#### 4. DATEFROMPARTS

## 5. DATENAME

## 6. DATEPART

## 7. DAY

## 8. GETDATE

## 9. GETUTCDATE

## 10. ISDATE

## 11. MONTH

## 12. SYSDATETIME

### 13. YEAR

#### 4. Advanced Functions:

## 1. CAST

## 2. COALESCE

### 3. CONVERT

#### 4. CURRENT\_USER

## 5. IIF

6. ISNULL
  7. ISNUMERIC
  8. NULLIF
  9. SESSION\_USER
  10. SESSIONPROPERTY
  11. SYSTEM\_USER
  12. USER\_NAME
- 

## MySQL Functions

### 1. String Functions:

1. ASCII  
Returns ASCII value of first character in string  
Ex: SELECT ASCII('HEMCHAND'); 72
2. CHAR\_LENGTH  
Returns the length of the string  
Ex: SELECT CHAR\_LENGTH('HEMCHAND'); 8
3. CHARACTER\_LENGTH  
Returns the length of the string  
Ex: SELECT CHARACTER\_LENGTH('HEMCHAND'); 8
4. CONCAT  
Concatenates the different strings  
Ex: SELECT CONCAT('HEM', 'CHAND');
5. CONCAT\_WS  
Concatenates the all the strings with the character we given  
Ex: SELECT CONCAT\_WS('-', 'HEM', 'CHAND', 'MOVVA'); HEM-CHAND-MOVVA
6. FIELD  
Returns the index of required val from the list of vals  
If there is no value or null it will return 0, case-insensitive search  
Ex: SELECT FIELD ('A', 'B', 'A', 'B'); 2

#### 7. FIELD\_IN\_SET

Ex: SELECT FIELD\_IN\_SET('A', "B,A,B"); 2

#### 8. FORMAT

Used to get the required decimal places, as well as for rounding (if no decimal pnt rqrd)

Ex: SELECT FORMAT(250500.5634, 2); 250500.56

Ex: SELECT FORMAT(250500.5634, 0); 250501

#### 9. INSERT

Inserting the new string in another string, for **replacing**.

Ex: SELECT INSERT('HEMCHAND', 1, 3, 'TEJA'); TEJACHAND

#### 10. INSTR

For searching the char

Ex: SELECT INSTR('HEMCHAND', E); 2

#### 11. LCASE

To convert text into lower case

Ex: SELECT LCASE('HEMCHAND') 'hemchand'

#### 12. LEFT

For getting the characters from the left side of string

Ex: SELECT LEFT('HEMCHAND', 3) 'HEM'

#### 13. LENGTH

Returns the length of the string

Ex: SELECT LENGTH('HEMCHAND'); 8

#### 14. LOCATE

For searching the character

Ex: SELECT LOCATE('3', 'W3SCHOOLS.COM'); 2

#### 15. LOWER

To convert text into lower case

Ex: SELECT LOWER('HEMCHAND'); 'hemchand'

#### 16. LPAD

Left padding the string with certain value/string by considering the total length of new string

Ex: SELECT LPAD('HEMCHAND', 10, '-'); '—HEMCHAND'

#### 17. LTRIM

Removes the leading spaces from the string

Ex: SELECT LTRIM(' HEMCHAND'); 'HEMCHAND'

## 18. MID

For extracting the **substring** from the string

Ex: SELECT MID('HEMCHAND', 4, 8); 'CHAND'

## 19. POSITION

To find index of substring

Ex: SELECT POSITION('HEM' IN 'HEMCHAND'); 1

## 20. REPEAT

To repeat the string for certain no of times

Ex: SELECT REPEAT("HEM", 3); 'HEMHEMHEM'

## 21. REPLACE

To replace all occurrences of certain substring with new one

Ex: SELECT REPLACE('JUICE', 'JU', 'N'); 'NICE'

## 22. REVERSE

To reverse the string

Ex: SELECT REVERSE('BHOLU'); 'ULOHB'

## 23. RIGHT

For getting the characters from right side of the string

Ex: SELECT RIGHT('HEMCHAND', 5); 'CHAND'

## 24. RPAD

Right padding the string with certain value/string by considering the total length of new string

Ex: SELECT RPAD('HEMCHAND', 10, '-'); 'HEMCHAND—'

## 25. RTRIM

Removes the trailing spaces from the string

Ex: SELECT RTRIM('HEMCHAND '); 'HEMCHAND'

## 26. SPACE

Just for creating the no of space

Ex: SELECT SPACE(10); ' ' ' ' ' ' ' ' ' ' ' ' ' '

## 27. STRCMP

Compares two strings. 0 when str1=str2, -1 when str1<str2, 1 when str1>str2

Ex: SELECT STRCMP('HEMCHAND', 'HEMCHAND'); 0

## 28. SUBSTR

To get substring

Ex: SELECT SUBSTR('HEMCHAND', 1, 3); 'HEM'

## 29. SUBSTRING

To get substring

Ex: SELECT SUBSTRING('HEMCHAND', 4, 8); 'CHAND'

## 30. SUBSTRING\_INDEX

Returns substring before specified no of delimiters applied

Ex: SELECT SUBSTRING\_INDEX('HELLO.WELCOME.TO.VVIT', '.', 1); 'HELLO'

## 31. TRIM

Removes leading the trailing spaces from string

Ex: SELECT TRIM(' HEMCHAND '); 'HEMCHAND'

## 32. UCASE

To convert text into upper case

Ex: SELECT UCASE('hemchand'); 'HEMCHAND'

## 33. UPPER

To convert text into upper case

Ex: SELECT UPPER('hemchand'); 'HEMCHAND'

## 2. Numeric Functions:

### 1. ABS

Converts neg value to pos

Ex: SELECT ABS(-10.5); 10.5

### 2. ACOS

Returns arc cosine of number

### 3. ASIN

Returns arc sine of number

### 4. ATAN

Returns arc tan of number

### 5. ATAN2

Return arc tan of two numbers

Ex: SELECT ATAN2(0.50, 1); 0.4636476090008061

### 6. AVG

Returns avg of values related to particular column

### 7. CEIL

Returns the small int value i.e greatest or equal to the number given

8. CEILING

Returns the small int value i.e greatest or equal to the number given

9. COS

Returns the cosine of number

10. COT

Returns the cotangent of number

11. COUNT

Returns the count of certain rows in table based on conditions

12. DEGREES

Converts radian values into degrees

13. DIV

Integer division

Ex: SELECT 10 DIV 5; 2

14. EXP

Returns the e to the power of certain number given

15. FLOOR

Returns the largest int i.e less than or equal to the number given number

16. GREATEST

Returns the greatest value of list of arguments

SELECT GREATEST(1, 4, 5, 2, 6); 6

SELECT GREATEST("ghh", "klk"); "klk"

17. LEAST

Returns the smallest value of list of arguments

18. LN

Returns the natural logarithm of given number

19. LOG

Returns the natural logarithm of given number

20. LOG10

Returns the base10 logarithm of given number

21. LOG2

Returns base2 logarithm of given number

22. MAX

23. MIN

#### 24. MOD

Returns the remainder

Ex: SELECT MOD(4, 2); 0

#### 25. PI

Returns the value of PI

Ex: SELECT PI();

#### 26. POW

Returns the power

Ex: SELECT POW(4, 2) 16

#### 27. POWER

Returns the power

Ex: SELECT POWER(4, 2) 16

#### 28. RADIANS

Converts the degrees into radians

#### 29. RAND

Returns the random number

Ex: SELECT RAND();

#### 30. ROUND

To round the number to specified decimal points

Ex: SELECT ROUND(148.909, 2); 148.91

Ex: SELECT ROUND(148.90, 0); 149

#### 31. SIGN

Returns the sign of number. 0 if num=0, 1 if num>0, -1 if num<0

#### 32. SIN

Returns the sin of the given number

#### 33. SQRT

Returns the sqrt of given number

#### 34. SUM

#### 35. TAN

Returns the tangent of given number

#### 36. TRUNCATE

SELECT TRUNCATE (148.909, 2); 148.90

### 3. Date Functions:

#### 1. ADDDATE

To add the time and date to particular date

Ex: SELECT ADDDATE(NOW(), **INTERVAL 10 DAY**);

Ex: SELECT ADDDATE("2024-04-10", INTERVAL 1 **MONTH**);

Ex: SELECT ADDDATE("2024/04/10", INTERVAL 1 **YEAR**);

Ex: SELECT ADDDATE("2024/04/10", INTERVAL -1 **YEAR**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **MICROSECOND**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **SECOND**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **MINUTE**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **HOURL**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **DAY**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **WEEK**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **MONTH**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **QUARTER**);

Ex: SELECT ADDDATE("2003-04-10", INTERVAL 10 **YEAR**);

#### 2. ADDTIME

To add time to the particular time or date

When working with times, it will helpful

days hh:mm:ss.ms

days hh:mm:ss.microsec

.

.

Hh:mm:ss

Ex: SELECT ADDTIME(NOW(), "02:00:00");

→adding two hrs

Ex: SELECT ADDTIME(NOW(), "00:00:00.000003");

→adding 3 micro secs

Ex: SELECT ADDTIME(NOW(), "4 20:34:35.000007");

→4days 20hrs 34mins 35sec 7mcro

Ex: SELECT ADDTIME("15:30:20", "00:30:59")

#### 3. CURDATE

Returns the current date

Ex: SELECT CURDATE();



#### 4. CURRENT\_DATE

Returns the current date

Ex: SELECT CURRENT\_DATE();

#### 5. CURRENT\_TIME

Returns the current time

Ex: SELECT CURRENT\_TIME();

#### 6. CURRENT\_TIMESTAMP

Returns the current date and time

Ex: SELECT CURRENT\_TIMESTAMP();

#### 7. CURTIME

Returns the current time

Ex: SELECT CURTIME();

#### 8. DATE

Extracts the date part

Ex: SELECT DATE(NOW());

Ex: SELECT DATE("2024-05-09 10:53:00") **2024-05-09**

#### 9. DATEDIFF

Returns the no of days between two dates

Ex: SELECT DATEDIFF(NOW(), "2023-10-05"); 336

#### 10. DATE\_ADD

Similar to ADDDATE

Ex: SELECT DATE\_ADD(NOW() INTERVAL 5 DAY);

#### 11. DATE\_FORMAT

To format the date specified

Syntax: DATE\_FORMAT(date, format)

##### Formats

%a Abbreviated weekday name (Sun to Sat)

%b Abbreviated month name (Jan to Dec)

%c Numeric month name(0 to 12)

%D Day of the month as numeric value followed by suffix (1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>..)

%d Day of the month as numeric value (01 to 31)

%e Day of the month as numeric value(0 to 31)

%f **Microseconds (000000 to 999999)**

<b>%H</b>	<b>Hour (00 to 23)</b>
<b>%h</b>	<b>Hour (00 to 12)</b>
<b>%l</b>	<b>Hour (00 to 12)</b>
<b>%i</b>	<b>Minutes (00 to 59)</b>
<b>%j</b>	Day of the year (001 to 366)
<b>%k</b>	Hour (0 to 23)
<b>%l</b>	Hour (1 to 12)
<b>%M</b>	Month name in full (January to December)
<b>%m</b>	Month name as numeric value(00 to 12)
<b>%p</b>	Pm or Am
<b>%r</b>	<b>Time in 12 hour AM or PM format (hh:mm:ss AM/PM)</b>
<b>%S</b>	<b>Seconds (00 to 59)</b>
<b>%s</b>	<b>Seconds (00 to 59)</b>
<b>%T</b>	<b>Time in 24hr format (hh:mm:ss)</b>
<b>%U</b>	Week where Sunday is the first day of week (00 to 53)
<b>%u</b>	Week where Monday is the first day of week (00 to 53)
<b>%V</b>	Week where Sunday is the first day of week (01 to 53) used with %X
<b>%v</b>	Week where Monday is the first day of week (01 to 53) used with %x
<b>%W</b>	Weekday name in full (Sunday to Saturday)
<b>%w</b>	Day of the week (0-Sunday, 1-Monday, 2-Tuesday, ..., 6-Saturday)
<b>%X</b>	Year for the week where Sunday is the first day of week. Used with %V
<b>%x</b>	Year for the week where Monday is the first day of week. Used with %v
<b>%Y</b>	Year as a numeric in YYYY format (4 digit value)
<b>%y</b>	Year as numeric in YY format (2 digit value)

Ex: SELECT DATE_FORMAT("2017-06-15", "%M %d %Y");	June 15 2017
Ex: SELECT DATE_FORMAT("2017-06-15", "%W %M %d %Y");	Thursday June 15 2017
Ex: SELECT DATE_FORMAT("2017-06-15", "%Y")	2017
Ex: SELECT DATE_FORMAT(	

## 12. DATE\_SUB

Just for subtracting the date and time

In DATEADD we will give negative values for subtracting

In DATE\_SUB we will give positive values for subtracting

Ex: SELECT DATE\_SUB(NOW(), INTERVAL 10 DAY)

### 13. DAY

Extracts the day of the month from given date

Ex: SELECT DAY(NOW()) 05

### 14. DAYNAME

Returns the week name of that day

Ex: SELECT DAYNAME(NOW()); Thursday

### 15. DAYOFMONTH

Equals to DAY

Ex: SELECT DAYOFMONTH(NOW()); 05

### 16. DAYOFWEEK

Returns the weekday index for a date

Ex: SELECT DAYOFWEEK(NOW()); 05(Thu) [1-sun,.....7-sat]

### 17. DAYOFYEAR

Returns the day of the year from 1 to 366

Ex: SEELCT DAYOFYEAR(NOW()); 249

### 18. EXTRACT

Extracts the part of the given date

Ex: SELECT EXTRACT(**MONTH** FROM NOW());

Ex: SELECT EXTRACT(**DAY** FROM NOW());

Ex: SELECT EXTRACT(**HOURL** FROM NOW());

Ex: SELECT EXTRACT(**MICROSECOND** FROM NOW());

Ex: SELECT EXTRACT(**YEAR** FROM NOW());

### 19. FROM\_DAYS

### 20. HOUR

Returns the hour part from given date (0 to 838)

Ex: SELECT HOUR(NOW());

### 21. LAST\_DAY

Returns the last day of the month

Ex: SELECT LAST\_DAY (NOW()); 2024-09-30

### 22. LOCALTIME

Returns current date and time

Ex: SELECT LOCALTIME();

### 23. LOCALTIMESTAMP

Returns current date and time

Ex: SELECT LOCALTIMESTAMP();

### 24. MAKEDATE

Create and return the date based on the year and the no of days value

Ex: SELECT MAKEDAT(2017, 200); 2017-07-19

### 25. MAKETIME

Create and return the time based on hh, mm, ss given

Ex: SELECT MAKETIME(04:34:23);

### 26. MICROSECOND

Returns the microseconds part of given date or time

Ex: SELECT MICROSECOND(NOW());

### 27. MINUTE

Ex: SELECT MINUTE(NOW());

### 28. MONTH

SELECT MONTH(NOW());

### 29. MONTHNAME

Returns the name of the month

Ex: SELECT MONTHNAME(NOW()); September

### 30. NOW

Returns the current date and time

Ex: SELECT NOW();

### 31. PERIOD\_ADD

### 32. PERIOD\_DIFF

### 33. QUARTER

Returns the quarter of the year of date

Jan to March 1

April to June 2

July to September 3

October to December 4

Ex: SELECT QUARTER(NOW()); 3

### 34. SECOND

Extracts and returns the second part of the datetime value

### 35. SEC\_TO\_TIME

Converts and returns the time based on the number of seconds

Ex: SELECT SEC\_TO\_TIME(3400); 00:56:40

### 36. STR\_TO\_DATE

Returns a date based on the string and format given

Ex: SELECT STR\_TO\_DATE("August 10 2017", "%M %d %Y"); 2017-08-10

### 37. SUBDATE

Subtract the date/time from the given date

Ex: SELECT SUBDATE(NOW(), INTERVAL 10 DAY);

### 38. SUBTIME

Subtract time from datetime/time expression

Ex: SELECT SUBTIME(NOW(), "3:2:5.000001");

Ex: SELECT SUBTIME("10:24:21", "5");

### 39. SYSDATE

Returns the current date and time

Ex: SELECT SYSDATE()

### 40. TIME

Extracts and returns the time part from given date or time

Ex: SELECT TIME(NOW());

### 41. TIME\_FORMAT

Formats a time by given format

Ex: SELECT TIME\_FORMAT("19:30:10", "%h %i %s %p"); 07 30 10 PM

Ex: SELECT TIME\_FORMAT("19:30:10", %r); 07:30:10 PM

### 42. TIME\_TO\_SEC

Converts the time value into seconds

Ex: SELECT TIME\_TO\_SEC ("19:30:10");

### 43. TIMEDIFF

Returns the difference between two time/datetime expressions

Given two param should be same format

Ex: SELECT TIMEDIFF(NOW(), "2013-10-10 12:34:40");

### 44. TIMESTAMP

Returns datetime value based on date/datetime

Ex: SELECT TIMESTAMP("2014:10:10"); 2014:10:10 00:00:00

#### 45. TO\_DAYS

Returns no of days between given date and 0

Ex: SELECT TO\_DAYS(NOW());

#### 46. WEEK (0 TO 53)

Returns the week number for given date

Ex: SELECT WEEK(NOW()); 35

#### 47. WEEKDAY

Returns weekday number for given date

0-Monday, 1-Tuesday,.....6-Sunday

Ex: SELECT WEEKDAY(NOW());

#### 48. WEEKOFYEAR (1 TO 53)

Returns the week number of date

Ex: SELECT WEEKOFYEAR(NOW()); 36

#### 49. YEAR

Returns the year part of given year

Ex: SELECT YEAR(NOW()); 2024

#### 50. YEARWEEK

Returns year and week number of given date

Ex: SELECT YEARWEEK("2017-10-25"); 201743

### 4. Advanced Functions:

#### 1. BIN

Returns the binary representation of the number in string format

Ex: SELECT BIN(10); 1010

#### 2. BINARY

#### 3. CASE

Go through the conditions, if condition met then returns a value

Ex: SELECT orderId, quantity,

CASE

WHEN quantity>30 THEN "Greater than 30"

WHEN quantity=30 THEN "Equal to 30"

ELSE "Less than 30"

END FROM ordersTable

#### 4. CAST

Convert a value of any type into another specified datatype

Values:

- DATE
- DATETIME
- DECIMAL Use optional M and D to specify max no of digits(M) and digits following decimal(D) point
- TIME
- CHAR
- NCHAR
- SIGNED
- UNSIGNED
- BINARY

Ex: SELECT CAST("2017-08-29" AS DATE);

str to date

Ex: SELECT CAST(32.0 AS **DECIMAL**(16, 6);

Ex: SELECT CAST(150 AS CHAR);

Ex: SELECT CAST("12:09:34" AS TIME);

#### 5. COALESCE

#### 6. CONNECTION\_ID

Returns the unique connection id for the current connection

Ex: SELECT CONNECTION\_ID();

#### 7. CONV

Converts a number from one numeric format to another

Ex: SELECT CONV(15, 10, 2)

1111

Ex SELECT CONV(1111, 2, 10)

15

#### 8. CONVERT

Similar to cast

Ex: SELECT CONVERT("12:54:33", TIME);

#### 9. CURRENT\_USER

Returns the username and hostname for MySQL account

Ex: SELECT CURRENT\_USER();

## 10. DATABASE

Returns the name of current database

Ex: SELECT DATABASE();

## 11. IF

Returns one value if condition met, another value otherwise

Es: SELECT IF(10>20, "BIG", "LESS");

## 12. IFNULL

Returns specified value if expression becomes null

Syntax: SELECT IFNULL(expression, returnValue)

Ex: SELECT IFNULL(bloodGroup, "Blood group not updated") FROM STUDENTS;

## 13. ISNULL

Returns 0 if expression is null, otherwise 1

Ex: SELECT ISNULL("HEL"); 1

## 14. LAST\_INSERT\_ID

Returns the Auto\_Increment Idof the last row that has been inserted in table

Ex: SELECT LAST\_INSERT\_ID(); 0

## 15. NULLIF

Compares the two expressions

Returns NULL if two expressions were same, returns first expression otherwise

Ex: SELECT NULLIF(23, 23) NULL

Ex: SELECT NULLIF(23, 27) 23

## 16. SESSION\_USER

Returns current user name and host name for MySQL connection

Ex: SELECT SESSION\_USER();

## 17. SYSTEM\_USER

Returns the current user name and host name for MySQL connection

Ex: SELECT SYSTEM\_USER();

## 18. USER

Returns the current user name and host name for MySQL connection

Ex: SELECT USER();

## 19. VERSION

Returns the current version of MySQL

Ex: SELECT VERSION();