FUNCTIONS IN DBMS

📌 String / Character Functions

Function	Syntax	Description	Example	Output
LOWER()	LOWER('TEXT')	Converts all characters to lowercase	SELECT LOWER('DBMS') FROM dual;	dbms
UPPER()	UPPER('text')	Converts all characters to uppercase	SELECT UPPER('dbms') FROM dual;	DBMS
INITCAP()	INITCAP('text')	Capitalizes the first letter of each word	SELECT INITCAP('open ai') FROM dual;	Open Ai
LENGTH()	LENGTH('text')	Returns the number of characters	SELECT LENGTH('Oracle') FROM dual;	6
SUBSTR()	SUBSTR('string', m, n)	Extracts substring from position m with length n	SUBSTR('SECURE', 3, 4)	CURE
INSTR()	INSTR('str', 'sub')	Finds position of substring	INSTR('DATABASE', 'BASE')	5
CONCAT()	CONCAT('a', 'b')	Combines two strings	CONCAT('Data', 'Base')	DataBase
ASCII()	ASCII('char')	Returns ASCII code of the first character	ASCII('A')	65
LTRIM()	LTRIM('abc', 'a')	Removes leading characters	LTRIM('nisha', 'n')	isha
RTRIM()	RTRIM('abc', 'c')	Removes trailing characters	RTRIM('sunila', 'a')	sunil
TRIM()	TRIM(' text ')	Trims leading/trailing	TRIM(' Hansel ')	Hansel

		whitespace or chars		
LPAD()	LPAD('str', n, 'x')	Left-pads a string to length n	LPAD('page1', 10, '*')	*****page1
RPAD()	RPAD('str', n, 'x')	Right-pads a string to length n	RPAD('ivan', 10, 'x')	ivanxxxxxx

---

🔢 Numeric / Arithmetic Functions

Function	Syntax	Description	Example	Output
ABS()	ABS(n)	Absolute value of n	ABS(-15)	15
POWER()	POWER(m, n)	m raised to power n	POWER(3, 2)	9
ROUND()	ROUND(n, m)	Rounds n to m decimal places	ROUND(15.91, 1)	15.9
TRUNC()	TRUNC(n, m)	Truncates n at m decimal places	TRUNC(15.91, 1)	15.9
MOD()	MOD(a, b)	Remainder of a/b	MOD(10, 3)	1
SQRT()	SQRT(n)	Square root of n	SQRT(25)	5
GREATEST()	GREATEST(a, b, c)	Returns the greatest value	GREATEST(4, 5, 17)	17
LEAST()	LEAST(a, b, c)	Returns the least value	LEAST(4, 5, 17)	4
CEIL()	CEIL(n)	Smallest integer ≥ n	CEIL(13.15)	14
FLOOR()	FLOOR(n)	Largest integer ≤ n	FLOOR(24.8)	24

🧮 Aggregate Functions

Function	Syntax	Description	Example	Output
AVG()	AVG(col)	Average of values	SELECT AVG(sal) FROM emp	25000
MIN()	MIN(col)	Minimum value	SELECT MIN(sal) FROM emp	20000
MAX()	MAX(col)	Maximum value	SELECT MAX(sal) FROM emp	120000

COUNT()	COUNT(col)	Number of non-null rows	COUNT(acct_no)	100
SUM()	SUM(col)	Total of values	SELECT SUM(sal) FROM emp	1350000

🗓️ Date Functions

Function	Syntax	Description	Example	Output
SYSDATE	SYSDATE	Current date and time	SELECT SYSDATE FROM DUAL	03-OCT-20
ADD_MONTHS()	ADD_MONTHS(d, n)	Adds n months to date d	ADD_MONTHS(SYSDATE, 4)	Feb-03-2021
MONTHS_BETWEEN()	MONTHS_BETWEEN(d1, d2)	Months between two dates	MONTHS_BETWEEN('25-DEC-81','25-DEC-79')	24
NEXT_DAY()	NEXT_DAY(date, 'DAY')	Next occurrence of a weekday	NEXT_DAY(SYSDATE, 'FRIDAY')	09-OCT-20
LAST_DAY()	LAST_DAY(date)	Last day of the month	LAST_DAY(SYSDATE)	30-SEP-13

🔁 Conversion Functions

Function	Syntax	Description	Example	Output
TO_CHAR()	TO_CHAR(n, 'fmt')	Converts number/date to string	TO_CHAR(SYSDATE, 'Mon')	Oct
TO_DATE()	TO_DATE(str, 'fmt')	Converts string to date	TO_DATE('03-10-2020','DD-MM-YYYY')	03-OCT-2020

TO_NUMBER()	TO_NUMBER('str')	Converts string to number	TO_NUMBER('123')	123

OPERATORS IN DBMS

➕ Arithmetic Operators

Operator	Description	Example	Output
+	Addition	5 + 3	8
-	Subtraction	5 - 3	2
*	Multiplication	5 * 3	15
/	Division	6 / 3	2

🔍 Comparison Operators

Operator	Description	Example	Output
=	Equal to	sal = 30000	True/False
!=, <>	Not equal to	dept <> 'HR'	True/False
<, >	Less/Greater than	sal > 25000	True/False
<=, >=	Less/Greater than or equal to	sal >= 25000	True/False

🔗 Logical Operators

Operator	Description	Example
AND	Both conditions must be true	sal > 30000 AND dept = 'IT'
OR	At least one condition must be true	dept = 'IT' OR dept = 'CSE'
NOT	Negates the condition	NOT sal < 30000

💡 Special Operators

Operator	Description	Example
LIKE	Pattern match (%, _)	FNM LIKE 'De__'
BETWEEN	Within range (inclusive)	per BETWEEN 50 AND 80
IN	Matches any in list	dept IN ('IT', 'HR')
NOT IN	Not in list	dept NOT IN ('IT', 'HR')
EXISTS	Returns true if subquery returns rows	WHERE EXISTS (SELECT * FROM ...)