



## SQL Functions Quick Reference

A practical cheat sheet for the most used SQL functions at work

by Data Desk Tools

*Published by Data Desk Tools. © 2026 Data Desk Tools. All rights reserved. Licensed for personal use only. Redistribution, resale, sharing, sublicensing, or uploading to marketplaces is not permitted.*

*Bonus file for SQL Query Toolkit*

Use this quick reference when you need to quickly remember which function to use for reporting, cleaning, finance analysis, sales analysis, Power BI preparation, or data quality checks.

## 1. Function Quick Lookup

Function group	Most used functions	Use when you need to	Common mistake
Aggregate functions	COUNT(), SUM(), AVG(), MIN(), MAX()	Use them for totals, counts, averages, maximum dates, minimum dates, and grouped business reporting.	Do not mix aggregated and non-aggregated columns without GROUP BY.
Text functions	TRIM(), UPPER(), LOWER(), LEFT(), RIGHT(), SUBSTRING(), REPLACE(), LENGTH(), LEN(), CONCAT()	Use them when imported data has spaces, inconsistent capitalization, long product codes, prefixes, suffixes, or values that need standardization before joins.	LEN() is common in SQL Server, while LENGTH() is common in Snowflake, PostgreSQL, MySQL, Oracle, and DB2.
Numeric functions	ROUND(), ABS(), CEIL(), FLOOR()	Use them for margin percentages, variance calculations, rounded KPI values, absolute differences, and report-ready numeric outputs.	When calculating percentages, protect divisions with NULLIF(value, 0) to avoid divide-by-zero errors.
NULL handling functions	COALESCE(), NULLIF(), ISNULL(), IFNULL()	Use them when reports show blanks, calculations return NULL, costs are missing, or percentages need safe division.	Do not automatically replace NULL with zero unless zero is really the correct business meaning.
Date functions	CURRENT_DATE, DATE_TRUNC(), EXTRACT(), YEAR(), MONTH(), DATEDIFF(), DATEADD()	Use them for monthly reports, YTD analysis, rolling periods, customer inactivity, aging reports, and period comparisons.	DATE_TRUNC() works in Snowflake/PostgreSQL. SQL Server, MySQL, Oracle, and DB2 may require different date syntax.
Conditional logic	CASE WHEN	Use it for customer segmentation, product status flags, risk labels, margin buckets, and report-friendly classifications.	Order your CASE WHEN conditions carefully. SQL stops at the first true condition.
Conversion functions	CAST(), TRY_CAST(), TO_DATE(), TO_NUMBER()	Use them when data comes from Excel, CSV, ERP exports, or legacy systems where dates and numbers are stored as text.	Check the original date format before converting. 01/02/2026 can mean different dates depending on locale.
Window functions	ROW_NUMBER(), RANK(), DENSE_RANK(), LAG(), LEAD()	Use them for latest price selection, deduplication, rankings, period comparisons, running totals, and moving averages.	Always define the correct PARTITION BY and ORDER BY. Without them, previous/next values and rankings can be misleading.

## 2. Practical Function Examples

### Aggregate functions

**Functions:** COUNT(), SUM(), AVG(), MIN(), MAX()

**What they do:** Aggregate functions summarize many rows into a single number, date, or value.

**When to use them:** Use them for totals, counts, averages, maximum dates, minimum dates, and grouped business reporting.

**SQL example:**

```
SELECT
customer_id,
COUNT(order_id) AS number_of_orders,
SUM(sales_amount) AS total_sales,
AVG(sales_amount) AS average_order_value,
MAX(order_date) AS last_order_date
FROM sales
GROUP BY customer_id;
```

**Sample result:**

customer_id	number_of_orders	total_sales	average_order_value	last_order_date
CUST001	12	18,450.00	1,537.50	2026-05-28
CUST002	7	9,200.00	1,314.29	2026-05-19

**Common mistake:** Do not mix aggregated and non-aggregated columns without GROUP BY.

## Text functions

**Functions:** TRIM(), UPPER(), LOWER(), LEFT(), RIGHT(), SUBSTRING(), REPLACE(), LENGTH(), LEN(), CONCAT()

**What they do:** Text functions clean, standardize, extract, and combine text values such as codes, names, emails, and categories.

**When to use them:** Use them when imported data has spaces, inconsistent capitalization, long product codes, prefixes, suffixes, or values that need standardization before joins.

**SQL example:**

```
SELECT
product_code,
LEFT(product_code, 3) AS product_family,
RIGHT(product_code, 4) AS product_suffix,
UPPER(TRIM(product_name)) AS product_name_clean
FROM products;
```

**Sample result:**

product_code	product_family	product_suffix	product_name_clean
ABC-1001	ABC	1001	INDUSTRIAL SENSOR
MTR-2045	MTR	2045	MOTOR CONTROLLER

**Common mistake:** LEN() is common in SQL Server, while LENGTH() is common in Snowflake, PostgreSQL, MySQL, Oracle, and DB2.

## Numeric functions

**Functions:** ROUND(), ABS(), CEIL(), FLOOR()

**What they do:** Numeric functions format, round, and normalize numeric values for analysis and reporting.

**When to use them:** Use them for margin percentages, variance calculations, rounded KPI values, absolute differences, and report-ready numeric outputs.

**SQL example:**

```
SELECT
product_id,
revenue,
cost_amount,
ROUND((revenue - cost_amount) / NULLIF(revenue, 0) * 100, 2) AS margin_percentage
```

```
FROM sales;
```

**Sample result:**

product_id	revenue	cost_amount	margin_percentage
P001	1,200.00	780.00	35.00
P002	950.00	700.00	26.32

**Common mistake:** When calculating percentages, protect divisions with NULLIF(value, 0) to avoid divide-by-zero errors.

## NULL handling functions

**Functions:** COALESCE(), NULLIF(), ISNULL(), IFNULL()

**What they do:** NULL handling functions replace missing values, prevent calculation errors, and avoid divide-by-zero problems.

**When to use them:** Use them when reports show blanks, calculations return NULL, costs are missing, or percentages need safe division.

**SQL example:**

```
SELECT
product_id,
unit_price,
unit_cost,
COALESCE(unit_cost, 0) AS unit_cost_clean
FROM products;
```

**Sample result:**

product_id	unit_price	unit_cost	unit_cost_clean
P001	120.00	80.00	80.00
P002	95.00	NULL	0.00

**Common mistake:** Do not automatically replace NULL with zero unless zero is really the correct business meaning.

## Date functions

**Functions:** CURRENT\_DATE, DATE\_TRUNC(), EXTRACT(), YEAR(), MONTH(), DATEDIFF(), DATEADD()

**What they do:** Date functions filter, group, compare, and calculate periods for reporting.

**When to use them:** Use them for monthly reports, YTD analysis, rolling periods, customer inactivity, aging reports, and period comparisons.

**SQL example:**

```
SELECT
DATE_TRUNC('month', order_date) AS sales_month,
SUM(sales_amount) AS total_sales
FROM sales
GROUP BY DATE_TRUNC('month', order_date)
ORDER BY sales_month;
```

**Sample result:**

sales_month	total_sales
2026-01-01	125,000.00
2026-02-01	138,500.00

**Common mistake:** DATE\_TRUNC() works in Snowflake/PostgreSQL. SQL Server, MySQL, Oracle, and DB2 may require different date syntax.

## Conditional logic

**Functions:** CASE WHEN

**What they do:** CASE WHEN creates business labels, flags, and categories directly inside a query.

**When to use them:** Use it for customer segmentation, product status flags, risk labels, margin buckets, and report-friendly classifications.

**SQL example:**

```
SELECT
  customer_id,
  total_sales,
  CASE
    WHEN total_sales >= 50000 THEN 'High Value'
    WHEN total_sales >= 10000 THEN 'Medium Value'
    ELSE 'Low Value'
  END AS customer_segment
FROM customer_sales;
```

**Sample result:**

customer_id	total_sales	customer_segment
CUST001	67,500.00	High Value
CUST002	18,500.00	Medium Value
CUST003	4,200.00	Low Value

**Common mistake:** Order your CASE WHEN conditions carefully. SQL stops at the first true condition.

## Conversion functions

**Functions:** CAST(), TRY\_CAST(), TO\_DATE(), TO\_NUMBER()

**What they do:** Conversion functions change values from one data type to another, such as text to date or text to number.

**When to use them:** Use them when data comes from Excel, CSV, ERP exports, or legacy systems where dates and numbers are stored as text.

**SQL example:**

```
SELECT
  order_id,
  order_date_text,
  CAST(order_date_text AS DATE) AS order_date
FROM orders;
```

**Sample result:**

order_id	order_date_text	order_date
SO-1001	2026-01-15	2026-01-15
SO-1002	2026-01-28	2026-01-28

**Common mistake:** Check the original date format before converting. 01/02/2026 can mean different dates depending on locale.

## Window functions

**Functions:** ROW\_NUMBER(), RANK(), DENSE\_RANK(), LAG(), LEAD()

**What they do:** Window functions calculate rankings, previous values, next values, latest records, and running calculations without collapsing rows like GROUP BY.

**When to use them:** Use them for latest price selection, deduplication, rankings, period comparisons, running totals, and moving averages.

#### SQL example:

```
SELECT
customer_id,
order_date,
sales_amount,
LAG(sales_amount) OVER (PARTITION BY customer_id ORDER BY order_date) AS previous_sales_amount
FROM sales;
```

#### Sample result:

customer_id	order_date	sales_amount	previous_sales_amount
CUST001	2026-01-10	1,200.00	NULL
CUST001	2026-02-12	1,450.00	1,200.00

**Common mistake:** Always define the correct PARTITION BY and ORDER BY. Without them, previous/next values and rankings can be misleading.

### 3. SQL Dialect Notes

SQL Server uses LEN() for text length. Snowflake, PostgreSQL, MySQL, Oracle, and DB2 usually use LENGTH().

SQL Server uses TOP 10. PostgreSQL, MySQL, and Snowflake commonly use LIMIT 10. Oracle and DB2 support FETCH FIRST 10 ROWS ONLY.

DATE\_TRUNC() is common in Snowflake and PostgreSQL. SQL Server, MySQL, Oracle, and DB2 may require different date syntax.

COALESCE() is widely supported. SQL Server also has ISNULL(), and MySQL has IFNULL().

TRY\_CAST() is useful where supported because it returns NULL instead of failing when conversion is not possible.

Always test date and conversion functions in your own SQL environment before using them in production.

### 4. Function Selection Checklist

Need a total, count, maximum, or average? Use aggregate functions such as SUM, COUNT, MAX, AVG.

Need to clean product codes, customer names, or emails? Use TRIM, UPPER, LOWER, LEFT, RIGHT, REPLACE.

Need to calculate percentages safely? Use ROUND and NULLIF.

Need to replace missing values? Use COALESCE, but only when the fallback value is business-correct.

Need monthly or yearly reporting? Use DATE\_TRUNC, EXTRACT, YEAR, MONTH, or database-specific date functions.

Need business labels or segments? Use CASE WHEN.

Need text-to-date or text-to-number conversion? Use CAST, TRY\_CAST, TO\_DATE, or TO\_NUMBER.

Need latest records, rankings, previous values, or deduplication? Use ROW\_NUMBER, RANK, LAG, or LEAD.