SQL Server vs MySQL vs PostgreSQL: Syntax Cheat Sheet

31 everyday tasks side by side. MySQL 8+ (most work in MariaDB), PostgreSQL 14+. SQL Server version notes in brackets.

Task	SQL Server	MySQL / MariaDB	PostgreSQL
SELECTING AND PAGING			
Top 5 rows	`SELECT TOP (5) ... ORDER BY x`	`... ORDER BY x LIMIT 5`	`... ORDER BY x LIMIT 5`
Page 3 (20 per page)	`ORDER BY x OFFSET 40 ROWS FETCH NEXT 20 ROWS ONLY`	`ORDER BY x LIMIT 20 OFFSET 40`	`ORDER BY x LIMIT 20 OFFSET 40`
Two-way IF	`IIF(cond, a, b)`	`IF(cond, a, b)`	`CASE WHEN cond THEN a ELSE b END`
Replace NULL	`COALESCE(x, 0) / ISNULL(x, 0)`	`COALESCE(x, 0) / IFNULL(x, 0)`	`COALESCE(x, 0)`
NULL-safe equal	`a IS NOT DISTINCT FROM b (2022+)`	`a <=> b`	`a IS NOT DISTINCT FROM b`
Case-insensitive LIKE	`LIKE (default collation)`	`LIKE (default collation)`	`ILIKE`
Regex match	`REGEXP_LIKE(s, 'p') (2025+)`	`s REGEXP 'p'`	`s ~ 'p'`
TEXT			
Join text	`CONCAT(a, b) or a + b`	`CONCAT(a, b)`	`a || b or concat(a, b)`
Length (characters)	`LEN(s)`	`CHAR_LENGTH(s)`	`length(s)`
Find position	`CHARINDEX('x', s)`	`LOCATE('x', s)`	`strpos(s, 'x')`
Part of a string	`SUBSTRING(s, 2, 3)`	`SUBSTRING(s, 2, 3)`	`substring(s, 2, 3)`
Rows to one list	`STRING_AGG(col, ', ')`	`GROUP_CONCAT(col SEPARATOR ', ')`	`string_agg(col, ', ')`
Split 'a,b,c' to rows	`STRING_SPLIT(s, ',')`	`JSON_TABLE(...) trick`	`string_to_table(s, ',')`
DATES			
Now	`SYSDATETIME() / GETDATE()`	`NOW()`	`now()`
Today (no time)	`CAST(GETDATE() AS date)`	`CURDATE()`	`current_date`
Add 7 days	`DATEADD(day, 7, d)`	`d + INTERVAL 7 DAY`	`d + INTERVAL '7 days'`
Days between	`DATEDIFF(day, d1, d2)`	`DATEDIFF(d2, d1)`	`d2 - d1`
Year of a date	`YEAR(d)`	`YEAR(d)`	`EXTRACT(YEAR FROM d)`
First day of month	`DATEFROMPARTS(YEAR(d), MONTH(d), 1)`	`DATE_FORMAT(d, '%Y-%m-01')`	`date_trunc('month', d)::date`
Last day of month	`EOMONTH(d)`	`LAST_DAY(d)`	`(date_trunc('month', d) + INTERVAL '1 month - 1 day')::date`
ISO week number	`DATEPART(iso_week, d)`	`WEEK(d, 3)`	`EXTRACT(WEEK FROM d)`
UTC to Bahrain time	`t AT TIME ZONE 'UTC' AT TIME ZONE 'Arab Standard Time'`	`CONVERT_TZ(t, '+00:00', '+03:00')`	`t AT TIME ZONE 'Asia/Bahrain'`
TABLES AND CHANGES			
Auto-number column	`INT IDENTITY(1,1)`	`INT AUTO_INCREMENT`	`INT GENERATED ALWAYS AS IDENTITY`
True / false column	`BIT`	`BOOLEAN (TINYINT(1))`	`boolean`
Unicode text	`NVARCHAR(100)`	`VARCHAR(100) (utf8mb4)`	`varchar(100) or text`
Get the new ID	`OUTPUT inserted.id / SCOPE_IDENTITY()`	`LAST_INSERT_ID()`	`RETURNING id`
Insert or update	`MERGE or UPDATE + IF @@ROWCOUNT = 0 INSERT`	`INSERT ... ON DUPLICATE KEY UPDATE`	`INSERT ... ON CONFLICT (key) DO UPDATE`
Replace a view	`CREATE OR ALTER VIEW v AS ...`	`CREATE OR REPLACE VIEW v AS ...`	`CREATE OR REPLACE VIEW v AS ...`
Run a procedure	`EXEC proc @p = 1`	`CALL proc(1)`	`CALL proc(1)`
See the query plan	`Ctrl+M in SSMS / SET STATISTICS IO ON`	`EXPLAIN ANALYZE SELECT ...`	`EXPLAIN ANALYZE SELECT ...`

Date difference order: SQL Server DATEDIFF(day, start, end) puts the start first; MySQL DATEDIFF(end, start) puts the end first.

Case sensitivity: SQL Server and MySQL compare text case-insensitively by default; PostgreSQL is case-sensitive, so use ILIKE or lower().

Unquoted names: PostgreSQL folds unquoted names to lowercase (Customers becomes customers). Use lowercase_with_underscores.