

Your SQL Doesn't Have to Be a Mess



Why Readability Matters

- Others (and future you) need to understand your logic
- Reduces time spent debugging and fixing errors
- Makes code reviews faster and more effective
- Facilitates onboarding new team members
- Readable SQL is often more performant SQL
- Builds trust in your data and analysis



Meaningful Names Are Key

- Use descriptive names for tables and columns
- Choose aliases that are short but meaningful
- Avoid vague names like "data" or "value1"
- Use consistent naming conventions (e.g., snake_case)
- Prefix columns with table aliases for clarity
- Name CTEs to describe their content clearly



Master the Art of Whitespace

- Use line breaks to separate logical sections
- Indent code within clauses like WHERE and FROM
- Align similar keywords vertically for easy scanning
- Avoid writing everything on one long line
- Use spaces around operators like = and +
- White space is free; use it generously to improve flow



Structure Your **SELECT** Clause

- List each column on its own line
- Align the AS keyword for aliases
- Put a comma at the front of each new line
- Include only the columns you truly need
- Group related columns together logically
- Comment on non-obvious calculations



Tame Your FROM and JOINS

- Always use explicit INNER JOIN syntax
- Use table aliases to shorten long table names
- Place the joined table and condition on the same line
- Indent JOIN conditions under the JOIN keyword
- List primary tables first, then lookups
- Use consistent aliases for the same table



Clarify Your WHERE Conditions

- List each condition on a new line
- Group related conditions with parentheses
- Indent conditions to show logical grouping
- Use comments for complex business logic
- Avoid overly complex nested conditions
- Place the most filtering conditions first



Leverage CTEs for Clarity

- Break complex logic into simple, named steps
- Use CTEs instead of deeply nested subqueries
- Each CTE should do one logical thing
- Name your CTEs descriptively (e.g., `user_events`)
- CTEs make your main query much simpler
- They act like chapters in a story



Comment With Purpose

- Explain the "why," not the "what"
- Comment on complex business rules or filters
- Note any temporary fixes or known issues
- Avoid stating the obvious (e.g., -- selects from users)
- Keep comments up-to-date with code changes
- Use single-line (--) for short notes



Consistent Formatting Style

- Choose a style guide and stick to it
- Be consistent with capitalization of keywords
- Standardize how you write common patterns
- This creates a familiar rhythm for readers
- Team-wide consistency is more important than personal preference
- Use automated formatters like SQLFluff



Test for Readability

- Step away from your code and return later
- Ask a colleague to review it for understanding
- Read it aloud to catch awkward phrasing
- Check if your formatting looks consistent
- Ensure your logic is easy to follow step-by-step
- The best query is one you can understand in six months



FOLLOW FOR MORE



Like this post? Repost!

Share your thoughts in the comments!

