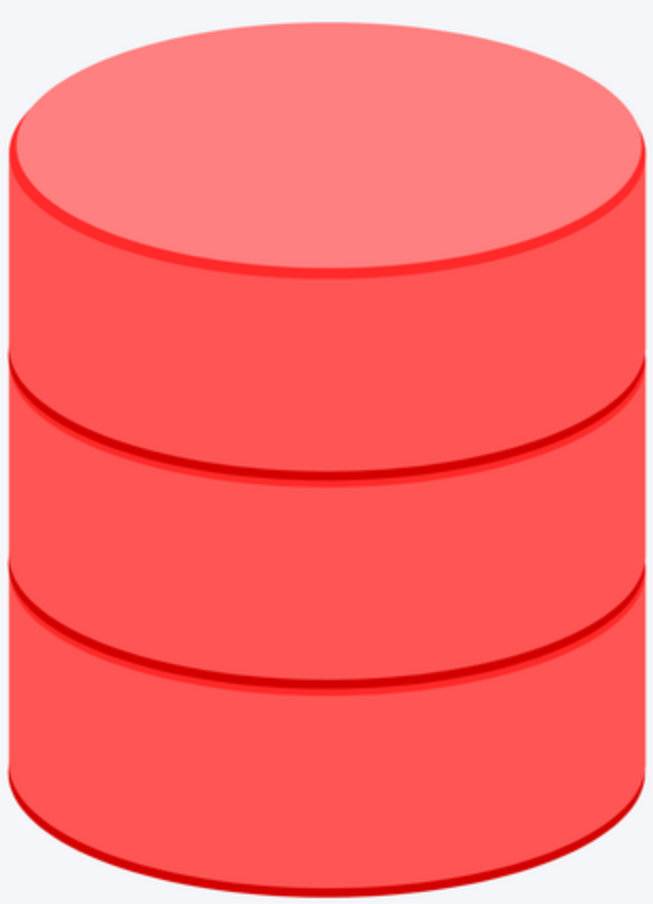
Your Data Workflow Is Costing You Time



The Full Journey

- Start with a raw SQL query
- Transform data for business logic
- Connect to a BI tool like Power BI
- Build visualizations and dashboards
- End with actionable insights
- Each step can be optimized

Why Reuse Queries?

- Saves significant development time
- Ensures consistency across reports
- Reduces errors from manual copying
- Simplifies maintenance and updates
- Makes onboarding new analysts easier
- Creates a single source of truth

Common Reuse Pitfalls

- Hard-coded values and filters
- Copy-pasting entire query blocks
- No central documentation
- Inconsistent naming conventions
- Missing comments for context
- One-off queries for similar needs

Modular Query Design

- Break complex logic into CTEs
- Use views for frequent logic blocks
- Create base queries for core metrics
- Build staging queries for transformations
- Layer queries for specific needs
- Reference other queries as building blocks

Parameterize Your Filters

- Replace hard-coded values with variables
- Use WHERE clauses with placeholders
- Define parameters for date ranges
- Allow dynamic selection of categories
- Enable user-driven filtering
- Parameters make queries flexible

BI Tool Parameters

- Power BI uses M language parameters
- Tableau has parameter controls
- Pass parameters to SQL queries
- Filter data at the source
- Improve performance by reducing data
- Create interactive user experiences

Performance: The Big Challenge

- BI tools are not databases
- They pull data into their engine
- Large datasets cause slow refreshes
- Complex joins can bottleneck
- Network latency adds to load times
- User interactions require quick responses

Push Work to the Database

- Let the database do the heavy lifting
- Filter and aggregate data in SQL
- Avoid importing raw, massive tables
- Use WHERE clauses to reduce rows
- Use GROUP BY to aggregate early
- Databases are built for this work

Optimize Your SQL

- Index columns used in JOINs and WHERE
- Avoid SELECT *; specify only needed columns
- Use efficient JOIN types
- Analyze query execution plans
- Break extremely complex queries into steps
- Consider materialized views for slow aggregates

Strategic Data Import

- Import aggregated data, not granular
- Schedule full refreshes during off-hours
- Use incremental refresh strategies
- Only refresh data that has changed
- Balance freshness with performance needs
- Cache data when appropriate

The Ideal Workflow

- Write a modular, parameterized SQL query
- Test performance in the database
- Connect to BI tool with parameters
- Import only the necessary data
- Build visuals on a lean dataset
- Share an interactive, fast dashboard

Your Action Plan

- Audit existing queries for reuse potential
- Identify common filters to parameterize
- Create a library of base queries
- Establish naming and documentation standards
- Test performance with sample parameters
- Train the team on the new approach

Key Takeaways

- Reusable queries save time and reduce errors
- Parameters enable flexibility and interactivity
- Always filter and aggregate at the source
- Database performance is dashboard performance
- A structured workflow is a efficient workflow
- Small optimizations compound into huge gains

FOLLOW FOR MORE

Like this post? Repost!

Share your thoughts in the comments!