

Your Data is a Mess. Let SQL Fix It.



Why Clean Data?

- Dirty data leads to inaccurate reports and analysis
- Ensures consistency across your entire dataset
- Builds a reliable foundation for machine learning models
- Saves time in the long run by preventing rework
- Essential for making confident business decisions
- Improves the performance of your database queries



Find Missing Values

- Use IS NULL to identify empty fields
- Use IS NOT NULL to find complete records
- Count nulls with COUNT(*) - COUNT(column_name)
- Use COALESCE to replace nulls with a default value
- Investigate why data is missing for critical fields
- Decide if rows with nulls should be kept or removed



Tame Text Data

- Use TRIM to remove extra spaces from strings
- Standardize case with UPPER() or LOWER() functions
- Replace substrings using the REPLACE function
- Check for inconsistent entries (e.g., 'USA', 'U.S.A.', 'US')
- Use LENGTH() to find abnormally long or short entries
- CONCAT strings to merge first and last names



Fix Data Types

- Ensure numbers are stored as INT or DECIMAL types
- Convert strings to proper DATE or DATETIME types
- Use CAST or CONVERT functions to change types
- Check for numbers mistakenly stored as text
- Validate that booleans are represented consistently
- Correct mismatched types before mathematical operations



Remove Duplicates

- Use GROUP BY and HAVING to find duplicate rows
- Identify duplicates based on key columns
- Use ROW_NUMBER() to rank and delete duplicates
- Keep the most recent or complete record
- Use SELECT DISTINCT for quick duplicate-free results
- Always validate counts before and after deduplication



Handle Outliers

- Use MIN() and MAX() to spot extreme values
- Calculate averages and standard deviations
- Filter data using WHERE and comparison operators
- Decide if outliers are errors or valid data points
- Use percentiles to identify top and bottom extremes
- Cap extreme values if they skew analysis



Standardize Formats

- Enforce consistent date formats (YYYY-MM-DD)
- Standardize phone numbers and country codes
- Apply a single currency format across financial data
- Create categories for free-text fields
- Use CASE statements to map values to a standard
- Ensure units of measurement are consistent



Validate Data

- Use CHECK constraints to prevent future errors
- Ensure foreign keys match primary keys in other tables
- Verify that numeric values fall within expected ranges
- Check that dates are logical (e.g., no future birth dates)
- Use LIKE with wildcards to find pattern violations
- Create data quality rules for ongoing checks



Your Cleaning Toolkit

- **UPDATE** to modify existing records
- **ALTER TABLE** to add, drop, or change columns
- **DELETE** to remove incorrect rows (use with caution)
- **CASE** statements for conditional logic and categorizing
- String functions (**TRIM**, **REPLACE**, **SUBSTRING**) for text
- Temporary tables to test changes before applying them



FOLLOW FOR MORE



Like this post? Repost!

Share your thoughts in the comments!

