# Naming Convention Best Practices (with PostgreSQL Considerations)

This table outlines best practices for naming database tables and columns, with PostgreSQL-specific recommendations to ensure clarity, consistency, and maintainability.

|  |  |  |
| --- | --- | --- |
| Aspect | Best Practice | PostgreSQL-Specific Guidance |
| Case Style | Use lowercase\_snake\_case for all object names. | PostgreSQL automatically folds unquoted names to lowercase. Using snake\_case avoids the need for quotes and improves consistency. |
| Word Separation | Use underscores \_ to separate words in table/column names. | Strongly recommended to avoid quoted identifiers and improve readability. |
| Table Names | Use singular nouns (e.g., user, order) or plural (e.g., users, orders) consistently. | Both are common in PostgreSQL. Plural is slightly more prevalent, but consistency is more important. |
| Column Names | Always use singular, descriptive names (e.g., first\_name, price). | Use snake\_case and avoid redundant table name prefixes (e.g., customer\_name in a customer table is redundant). |
| Primary Keys | Use id or [table]\_id as the primary key. Pick one convention and stick to it. | Both are supported. [table]\_id is often clearer in joins and foreign keys. |
| Foreign Keys | Name using [referenced\_table]\_id (e.g., user\_id, product\_id). | Recommended for clarity and self-documenting schema. |
| Avoid Reserved Words | Never use SQL keywords as names (e.g., user, select). | PostgreSQL requires quoting reserved words; best to avoid them altogether. Use SELECT pg\_get\_keywords(); to check reserved terms. |
| Special Characters | Avoid spaces, dashes, and special characters. Stick to letters, numbers, and underscores. | Special characters require quotes and increase complexity in PostgreSQL. |
| Prefixes/Suffixes | Avoid data-type prefixes like str\_, int\_, or suffixes like \_dt. | Some exceptions for object types: v\_ for views, mv\_ for materialized views, tmp\_ for temp tables are acceptable. |
| Boolean Columns | Use is\_, has\_, or can\_ prefixes (e.g., is\_active, has\_permission). | Widely supported and aligns with PostgreSQL readability conventions. |
| Timestamp Columns | Use \_at or \_on suffixes to clarify intent (e.g., created\_at, delivery\_on). | Consistent naming helps when using time-based triggers, audit logs, etc. |
| Index/Constraint Names | Follow a structured format (e.g., PK\_users, FK\_orders\_users, IX\_products\_category). | Helps identify object type and scope easily in PostgreSQL system catalogs. |
| Length Limit | Keep names under 63 characters. | PostgreSQL enforces a 63-character limit on identifiers. Names longer than this are truncated internally. |
| Schema Names | Use lowercase and meaningful names (e.g., inventory, analytics). | Apply same naming rules as tables. Schemas are ideal for logical grouping of tables and permissions. |
| Documentation | Maintain a shared naming convention document (README, wiki, or ERD notes). | Strongly encouraged for team alignment. PostgreSQL does not enforce naming rules—discipline must come from process and documentation. |