

[<< Back to Dashboard](#)

Database Naming Conventions

Contents

- [Database Naming Conventions](#)
 - [General Conventions](#)
 - [Table Conventions](#)
 - [Column Conventions](#)
 - [Index Conventions](#)
 - [Views Convention](#)
 - [Related Guides](#)

A naming convention is important when working with databases in order to provide a clear structure and format; especially in team environments. By using these conventions the database can be understood by anyone applying these conventions; thus increasing maintainability. These are not rules but guidelines that can be adapted to any working environment.

General Conventions

- All names should be in camelcase with the first letter in lower case.

```
-- DO THIS --
users
permissions
userPermissions
```

```
-- NOT THIS --
Users
Permissions
User_Permissions
user_permissions
```

- Separate name parts by using camel case and NOT underscores or spaces. This provides better readability and you will not have to use quotes when doing SQL statements. Look at example in the previous point.
- Prefixes or namespaces are the ONLY parts of names that should be separated by underscores. This defines a clear separation between names and areas.

```
-- DO THIS --
blog_users
blog_userPermissions
```

```
-- NOT THIS --
BlogUsers
BlogUserPermissions
Blog_UserPermissions
```

- Do not use numbers in names. This is poor design, indicating divided table structures.
- Do not use dot (.) separator between names, remember use camel casing. This way you will avoid problems when doing SQL statements as fields are accessed using dot notation.
- It goes without saying, do not use reserved database words in any name.
- Always try to use names that make sense and are descriptive of their purpose.
- Avoid abbreviations whenever possible. ONLY use abbreviations that are well known and documented.
- Avoid acronyms whenever possible. Only use acronyms that are well known and documented. Also, they should all be in uppercase if used.

```
-- DO THIS --  
HTTPRequest  
savedURL
```

```
-- NOT THIS --  
http_requests  
httprequests  
savedurl  
saved_url
```

Table Conventions

- Tables are usually entity you are modeling for persistence. So make sure the names are in proper English and carry natural meanings. These names should make sense and should be descriptive.
- Avoid acronyms and abbreviations if at all possible. If acronyms are used, then make sure to capitalize them. Abbreviations used should be well known abbreviations and should be camel cased.
- Avoid using plural names for tables, use singular forms. This will avoid errors due to pluralization of words and also when moving table design to objects or entities. The name should map to the entity it is modeling.

```
-- DO THIS --  
blog_user  
blog_page  
person  
box  
activity
```

```
-- NOT THIS --  
blog_users  
blog_pages  
people  
boxes  
activities
```

- Use namespaces for tables whenever grouping is needed. This grouping is a clear separation when working with the tables and can also provide a good basis for security, as you could secure via the namespace. Namespaces should be separated by an underscore and then the name should be camel cased.

```
-- DO THIS --
lookups_countries
lookups_states
lookups_regionalOffices
hr_employees
hr_salaries
is_employees
is_vacationDays
```

```
-- NOT THIS --
LookupsCountries
lookupsStates
LOOKUPS_RegionalOffices
HREmployees
ISEmployees
ISVacationDays
```

- Do not use prefixes like *tbl,db* as they are EXTREMELY redundant and useless.
- All tables should have at least ONE primary key.

Column Conventions

- Columns are usually attributes of an entity you are modeling via a table. So make sure the names are in proper English and carry natural meanings. These names should make sense and should be descriptive.
- Avoid acronyms and abbreviations if at all possible. If acronyms are used, then make sure to capitalize them. Abbreviations used should be well known abbreviations and should be camel cased.
- Get used to naming primary or foreign keys with an *ID* identifier suffix. DO NOT use the column name *id* to identify a primary key. Be more descriptive, usually looking at the table name.

```
-- DO THIS --
userID
userPermissionID
```

```
-- NOT THIS --
id
userid
upid
```

- Foreign keys should be prefixed with a capitalized *FK_* in order to visually denote relationships and also to avoid ambiguous table selects. After the

FK_ the name of the column must comply with our naming conventions.

```
-- DO THIS --  
FK_userID
```

```
-- NOT THIS --  
fkuserid  
userID
```

- Boolean columns should use the *is* prefix.

```
-- DO THIS --  
isActive  
isSold
```

```
-- NOT THIS --  
active  
sold
```

- Date columns should be suffixed in camel case with the word *Date*.

```
-- DO THIS --  
createdDate  
updatedDate
```

```
-- NOT THIS --  
date_created  
date_updated
```

- Try to make

Index Conventions


- Always prefix all index names with *idx_* to distinguish it as an index.
- The names of the index should at least convey the columns it reaches or at least the purpose of the index.

Views Convention

- Always try to use names that make sense and are descriptive for the views. Avoid acronyms and abbreviations if at all possible. If acronyms are used, then make sure to capitalize them. Abbreviations used should be well known abbreviations and should be camel cased.
- All views should be prefixed with *vw_* in order to distinguish them in SQL statements from regular database tables.

Related Guides

- [Model-View-Controller Demystified](#)
- [ColdFusion Development Best Practices](#)
- [ColdFusion SQL Injection Protection Best Practices](#)

 Categories:

- [theory](#)