Taken from here.

Oracle Naming Conventions

Nothing is carved in stone, but these are the sort of rules we follow:

- Entities & Tables
- Aliasing
- Attributes & Columns
- Keys & Their Columns
- Indexes
- Triggers
- Other Objects
- PL/SQL Variables
- File Extensions

Entities & Tables

All entity names should be singular and may have spaces. These are replaced with '_' by Oracle Designer during table creation.

```
APPLICATION
APPLICATION FUNCTION
APPLICATION FUNCTION ROLE
```

All table names should be plural. If the table name contains serveral words, only the last one should be plural.

```
APPLICATIONS
APPLICATION_FUNCTIONS
APPLICATION_FUNCTION_ROLES
```

Aliasing

All entities and tables should be defined a unique alias that is used as part of the foreign key definitions. The alias should be an abbreviation of the name.

```
APPLICATIONS = APPL (4)
APPLICATION_FUNCTIONS = APFU (2:2)
APPLICATION_FUNCTION_ROLES = APFR (2:1:1)
APPLICATION_FUNCTION_ROLE_BANANAS = AFRB (1:1:1:1)
APPLICATION_FUNCTION_ROLE_BANANA_APPLES = (Do what you like!)
```

The bracketed numbers indicate the number of letters used from each word in the name.

Oracle has a 30 character restriction on object names, so table aliases are useful to reduce the length of object names.

Attributes & Columns

Attributes and columns should not be prefixed with a table alias. For the most part this is unnecessary and often very messy.

Keys & Their Columns

Primary Keys are named after the table or its alias with the suffix of '_PK'.

```
Table      : APPLICATIONS
Primary Key: APPLICATIONS_PK
             or
             APPL_PK
```

Unique Keys are named after the table or its alias with the suffix of '_UK'. If more than one unique key is present you may need to add further information to make the key name unique.

```
Table      : APPLICATIONS
Unique Key : APPLICATIONS_UK
             or
             APPL_UK
```

Foreign Keys are usually named using the two table aliases, from and to, with the suffix of '_FK':

```
Relationship: APPLICATION_FUNCTIONS -> APPLICATIONS
Foreign Key : APFU_APPL_FK
```

The foreign key column on the dependant table may be named using the full table name or the alias, along with the column name.

```
APPLICATION_FUNCTIONS.APPLICATION_ID -> APPLICATIONS.ID
or
APPLICATION_FUNCTIONS.APPL_ID -> APPLICATIONS.ID
```

I prefer the full table name, but the the 30 character limit forces the use of the alias at times.

Indexes

Indexes are created implicitly to support PKs and UKs. These indexes have the same name as the constraint.

As a rule of thumb, all foreign key columns should be indexed. Indexes to support foreign keys should be named using the foreign key name with the suffix '_I'.

```
Foreign Key    : APFU_APPL_FK
Suporting Index: APFU_APPL_FK_I
```

Other indexes should be created with meaningful names, usually incorporating the table alias and the column name(s) where possible, along with the suffix '_I'.

Triggers

Trigger names should be made up of the table name, an acronym representing the triggering action and the suffix "_TRG".

```
Table : APPLICATIONS

Action: BEFORE INSERT STATEMENT-LEVEL
Name  : APPLICATION_BIS_TRG

Action: AFTER INSERT AND UPDATE ROW-LEVEL
Name  : APPLICATION_AIUR_TRG
```

Other Objects

Other database objects have a suffix that identifies their object type.

```
Check Constraints : <name>_CHK
Sequences         : <name>_SEQ
Views             : <name>_V
Materialized Views: <name>_MV
Types             : <name>_T
Directories       : <name>_DIR
External Tables   : <name>_EXT
PL/SQL Packages   : <name>_API
PL/SQL Procedures : <name>_PRC
PL/SQL Functions  : <name>_FUN
```

PL/SQL Variables

PL/SQL variables are prefixed with a single letter, if possible, to indiate their type or usage.

```
Package Global Variables: g_variable_name
Local Variables         : l_variable_name
Types                   : t_type_name
Cursors                 : c_cursor_name
Exceptions              : e_exception_name
Input Parameters        : i_parameter_name
Outut Parameters        : o_parameter_name
In/Out Parameters       : io_parameter_name
```

Sometimes I flip back to using a generic "p_" prefix for parameters, regardless of their IN/OUT usage. Old habits dies hard.

File Extensions

File extensions include:

```
.pks – Package specification.
.pkb – Package body.
.sql – Everything else.
```