DB Design Standards (Version 2020.1)

# Architectural Guide lines

This document provides details about the data structure on the Dhansol databases.

System

User

Functional

# Naming Convention

The first two bytes will indicate the object type.

1. All the Database tables will begin with a prefix of “t”. with the lower-case.
2. All the database views will begin with a prefix of “v”. with lower case.
3. All the database stored procedures will begin with a prefix of “s” with lower case.
4. All the database materialized views will begin with a prefix of “m” with lower case.
5. All the queries will begin with a prefix of “q” with lower case.
6. The second byte will indicate the whether it is a system, user or functional table. It will carry ‘s’ for system, ‘u’ for the User specific tables and ‘f’ for the functional specific database objects.
   1. ‘s’ – System
   2. ‘u’ – User
   3. ‘f’ – Functional
7. Each of the naming prefix will be separated by “\_” except for the second byte where db object’s purpose domain type is specified which are System, User and Functional specified.
8. 4th Byte will indicate whether the object belongs to Application or Module.
   1. A – Core Application Data (Functional)
   2. M – Module (Calendar, Email, Provider, Practitioner etc.) – Independent Modules that can be added and subtracted.
   3. C – Custom set of tables specific for implementations.
9. 5th byte will indicate whether the object is a Application System, Application Content table or Application Reference table.
   1. “\_AC\_” – Application Content
   2. “\_AS\_” – Application Internal System
   3. “\_MC\_” - Module Content
   4. “\_AR\_” – Application Reference
   5. “\_MR\_” – Module Reference
10. from the 7th byte the name will begin the Module name and data category. The data category will begin with the camel case such as “Provider”. Where data category is unusually longer, abbreviation will be used to indicate the data category.

Ex. “tf\_AC\_Prov\_Contact”, “tf\_AC\_Coll\_Collaborations”

# Reference

Module or Application Content Reference will carry “Ref” suffix at the end.

Ex. “tf\_MC\_FeeSched\_TypeRef”

# Keys & ID Generation

Primary Keys for the reference and entity tables will be created by Surrogate keys, which will be a unique sequence created by the database sequence. For the relationship tables reference and entity table keys will be used, except at places where the hierarchy goes beyond more than 4 levels, which case a new Surrogate key will be created for the relationship keys.

Each entity and relationship tables will have effective periods.

Database sequence object will be used sequence generator.

Primary key will be a 8 byte long numeric data type as such as Double to accommodate a longest length.

# Transactions

CRUD operations will use transactions which are controlled at the applications transaction coordinator level.

# History

Each reference and entity table can be stored in the history table, with the application level installation setup option.

# User Session cache key

User Session Cache key will be created with a GUID object.

# Abbreviations