

Oracle Database Development Standards For DNR Staff and Contractors

Table of Contents

| | |
|--|---|
| INTRODUCTION..... | 2 |
| DATABASE ORGANIZATION | 2 |
| DATABASE PROCEDURES..... | 3 |
| Development | 3 |
| Testing..... | 3 |
| Production Release..... | 4 |
| New Schemas | 4 |
| Existing Schemas | 4 |
| CERTIFICATION and DOCUMENTATION..... | 5 |
| DATABASE APPLICATION DEVELOPMENT FLOW CHART..... | 6 |
| NAMING CONVENTIONS..... | 7 |

INTRODUCTION

The purpose of this document is to describe the managed-database policy for the Oracle database environment at DNR, along with naming convention requirements and process controls for DNR Programmers and Contractors. It is designed to clarify the Policies and procedures for database development.

DATABASE ORGANIZATION

The DNR Oracle database is organized to support industry best practices. This infrastructure isolates development and test activity from production databases. The table below defines the terms used in the rest of this document and explains the impact for DNR's particular database installation:

| | |
|--------------------------|---|
| Development Staff | DNR Programmers and Contractors assigned to the database development project. |
| INSTANCES | Autonomous database environments which can be managed completely independently from one another. Each instance fulfills a specific role in the development of software for Oracle databases. |
| DNRDEV | A development environment. All schema creation and modification occurs here. Application testing for newly-created or modified schemas must use this instance. Only upon successful alpha-testing should an application schema be duplicated in the test instance. |
| DNRTEST | A simulated production environment including database security as it will be in the production database. Data bulk-loading can occur here, but schema alteration cannot. This instance is used for application schema beta-testing, including compatibility with applications which already exist in the production environment. |
| DNRPROD | A production environment. Data bulk-loading can occur here, but schema alteration cannot. |
| Roles | Oracle provides for easy and controlled privilege management through roles. Roles are named groups of related database privileges that are granted to Oracle user ids or other roles. Each application schema receives two basic roles by default in its creation. These roles contain the database access privileges to an application schema's objects. |
| [schema]_users | Grants read-only access to Oracle objects within the schema. |
| [schema]_editor | Provides Select, Insert, Update and Delete privileges on Oracle objects within the schema. |
| User IDs | IDs that provide application access to database objects |
| Schema-id | An ID used to maintain an application schema (owner). Lead developers are given schema-owner-level access for the creation or removal of Oracle objects in the schema. Schema-ids are only open for login in the development instance(s) and are not to be used within client applications. These are locked in DNRTEST and DNRPROD. |
| Application-id | An Oracle login ID assigned to an application which provides access to all required Oracle database objects. This ID is intended for internal use by an application to access the Oracle backend data store. There are typically two ID's assigned to an application, read-only and data editing. |
| Login-id | An id assigned by a developer to an end-user for the purposes of logging into an application and establishing privileges. This id does not provide direct access to the database, only access through the application(s) designed to accept it. This has the positive effect of minimizing the number of user-ids with direct access to the database. |
| Alpha Testing | Testing carried out by the Development staff during application development. |
| Beta Testing | Final application testing carried out by select end users and Development staff. |

DATABASE PROCEDURES

The DNR Oracle database procedures are designed to support a documented, repeatable, defined, and managed process for database development.

Contractors may develop database schemas and their applications at the contractors shop and deliver their database product via an Oracle schema export. In this option, application testing starts in the development instance (DNRDEV). When permitted by DNR management contractors may also do development work in-house at DNR. Based on the nature of the work to be accomplished, the DBA and DNR Project Manager will assign the appropriate instance for Contractor work. Both options will adhere to the same standards and procedures required of DNR development staff. These basic requirements and procedures apply:

- 1) For new application schemas, acquire an approved schema name from the DBA (DNR Oracle Database Administrator) prior to database development.
- 2) Database designs both logical and physical are to be reviewed by the DBA and DNR project development staff prior to any application development begins. An ER diagram must be provided for this review; on request the DBA can provide an ERwin ER Diagram. Partial designs can be submitted when application development is managed in phases.
- 3) Applications will not use schema owner Oracle User IDs to initiate communications with the database for production runtime operations. Applications are provided two Oracle User IDs for their internal use to connect to the database, one read-only and the other for data editing. Applications are expected to manage end user Login-ids when applicable.
- 4) Before moving a schema into DNRTEST, basic database documentation must be completed and comments completed on tables, views and their attributes.
- 5) Before moving a schema into DNRPROD, all database documentation all must be completed before release to production.

Please note that advance planning with the DBA is desired.

Development

Application developers assigned to a project identify the need for a new application schema or modification to an existing schema. The lead developer works with the DBA to establish and configure the new schema as required, including any special application-ids which might be required for the project. When requested, the DBA can assist in the development of the data model and physical database objects. Once created, developers assigned to the project have DBA-level privileges to the new schema and are free to engage in whatever activities are required. These activities only take place in the development database instance DNRDEV. Both forms of testing alpha and beta can be carried out in DNRDEV.

Testing

When development staff have completed all application development in DNRDEV the DBA will copy all affected database elements (schemas, roles, user-ids) to DNRTEST. Beta-testing for the new (or modified) application and schema is then carried out in this simulated production environment by the development team and by designated end users before the schema is released into production.

When beta-testing has revealed the need for schema modifications, development staff must make those modifications in DNRDEV. When the modifications have been tested in DNRDEV and is again ready for DNRTEST, a new iteration to DNRTEST is executed by the DBA. The previous iteration is deleted in DNRTEST and a new copy is installed from the modified DNRDEV application schema, and beta-testing and regression-testing begins anew. This process continues until the development staff is ready to move the new or modified schema into the production environment.

Production Release

Schemas are placed into database instance DNRPROD by the DBA once beta testing has been concluded.

New Schemas

New application schemas can be installed from DNRDEV or DNRTEST with or without data (developer's option). This option allows new data, separate from test data, to be bulk loaded for initial production start. A copy is taken from the source (DNRDEV or DNRTEST). This copy is then applied to the target database instance DNRPROD.

Existing Schemas

When upgrading schemas, the DBA will use Oracle tools and or manual methods to reconcile the differences between DNRDEV and DNRPROD versions. These tools produce scripts that can be used to apply the changes to the target database instance DNRPROD. These scripts are validated in DNRTEST before applying them in DNRPROD.

The validation process executed by the DNR DBA is as follows. This procedure is a validation of the methodology and scripts used to move the schema into production.

- 1) Insure database schemas are the same in both DNRDEV and DNRTEST
- 2) Delete the database schema from DNRTEST.
- 3) Copy from DNRPROD to DNRTEST the current production version of the schema including all data.
- 4) Reconcile schema differences between DNRDEV and DNRTEST creating reconcile SQL scripts
- 5) Apply reconcile scripts in DNRTEST.
- 6) Lead Developer initiates beta level application testing in DNRTEST.
- 7) After development staff is convinced the application operates without error in DNRTEST, the lead developer requests movement into DNRPROD.
- 8) Freeze data updates for the schema in DNRPROD
- 9) Apply reconcile scripts in DNRPROD.
- 10) Development staff deploys new application version to production
- 11) Release application schema for data updates.

If beta testing reveals the need for schema changes, these changes must be done in DNRDEV. Then repeat steps 1 – 6 as necessary.

CERTIFICATION and DOCUMENTATION

To qualify a new or modified schema for production status, the development staff must:

- Request the new or modified schema move to DNRTEST for beta testing
- Perform due diligence with respect the schema's compatibility with existing applications and or other application schemas.

This means the development staff will do one of the following:

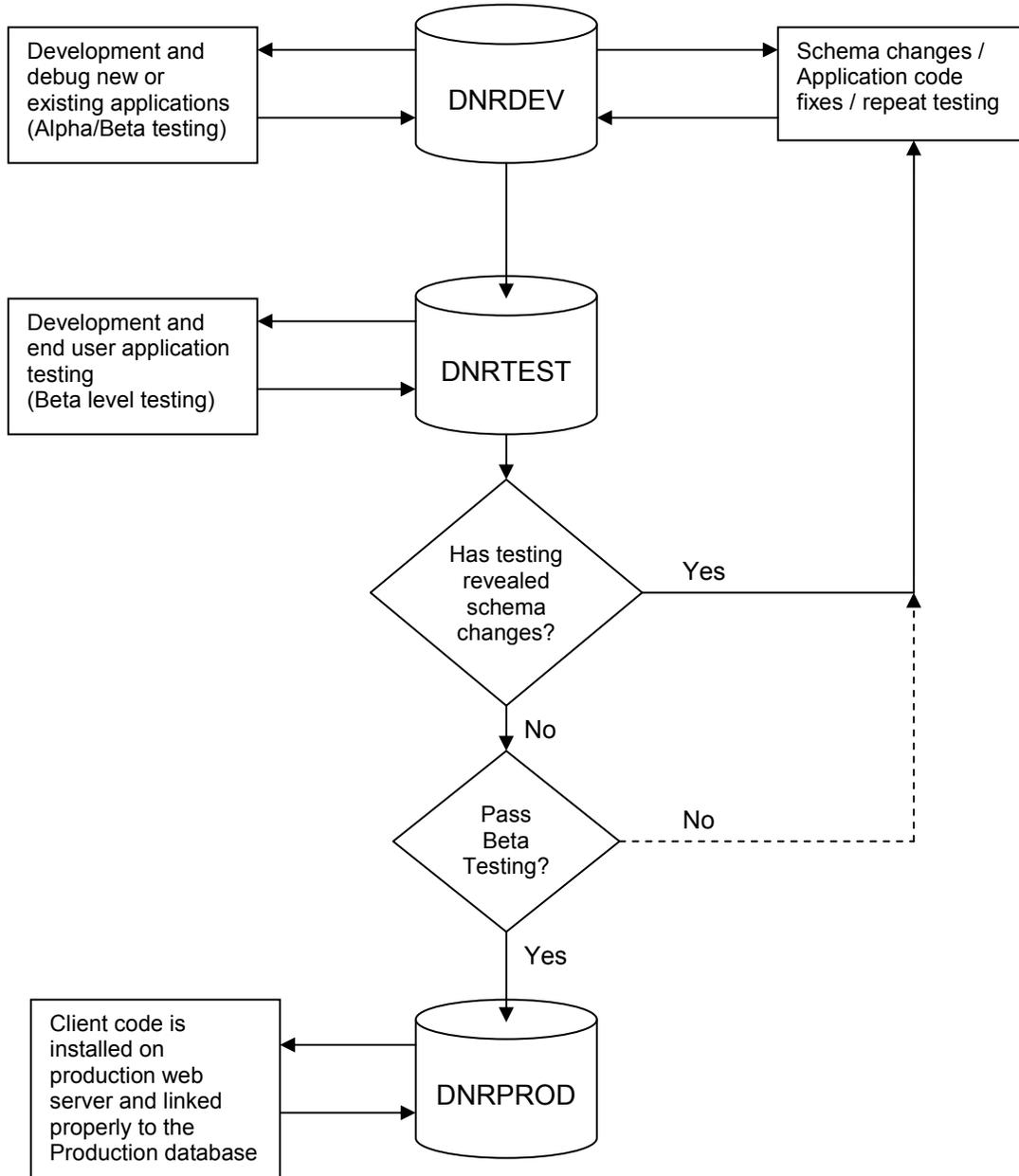
- ✓ Certify the schema is new having no dependences on any other schema, thus requiring no testing with existing applications.
 - ✓ Identify external dependencies and conduct and certify compatibility testing with dependent applications.
 - ✓ Identify external dependencies and work with the appropriate staff to schedule, conduct, and certify compatibility testing with dependent applications.
- Provide a minimal suite of documentation to be included in the DBA's production environment documentation binder and web site.

Certification of fitness for production status and regression testing performance consists of a cover sheet attached to the documentation suite, which must at a minimum include the following:

- 1) A brief description of the application's purpose describing both its business function and user community.
- 2) ER Diagram of the physical database schema; on request the DBA can provide an ERwin ER Diagram and related documentation during development.
- 3) Meaningful and adequate comments added to each table and its attributes using "comments on" in their DDL (Data Definition Language) SQL.
- 4) Suitable brief textual descriptions of all other schema objects.
- 5) Larger projects will include a reference to the project's design documents and product web site.

Upon acceptance of the documentation suite, the DBA will work with the lead developer to place the schema into the production instance.

DATABASE APPLICATION DEVELOPMENT FLOW CHART



NAMING CONVENTIONS

When creating database objects the following naming conventions apply. Database applications must use the fully qualified name in application code, `schema_name.object_name`.

| Object name | Convention | Description |
|--------------------|-------------------------------------|--|
| Table | none | The developer is free to use appropriate table names that are suggestive of its contents. No special characters allowed, only Letters, numbers and underscores |
| View | V_[view_name] | Begin all views with v_ |
| Materialized Views | MV_[view_name] | Begin all materialized views with MV_ |
| Primary Keys | PK_[table_name] ¹ | Begin a tables primary key name with PK_ |
| Foreign Keys | FK(1..n)_[table_name] ¹ | Tables with a foreign key should begin with FK. Tables with more than one foreign key should use FK1_, FK2_ and so on for each |
| Indexes | IDX(1..n)_[table_name] ¹ | Table indexes are to begin with IDX_. Tables with more than one index are to use IDX1 , IDX2 and so on for each |
| Sequences | SEQ(1..n)_{name} | Sequences are to begin with SEQ_. The developer is free to use appropriate sequence names that are suggestive of its role. |
| Triggers | TRG(1..n)_[table_name] ¹ | Triggers are to begin with TRG_. Tables with more than one trigger are to use TRG1 , TRG2 and so on for each |
| Packages | PKG_{name} | The developer is free to use appropriate package names that are suggestive of its contents. |
| Procedures | PRC_{name} | The developer is free to use appropriate procedure names that are suggestive of its contents. |
| Functions | FCN_{name} | The developer is free to use appropriate function names that are suggestive of its contents. |

¹ Name of the table the object resides