Corporate Services for the Natural Resource Sector

Information Management Branch

NRS Standards for Data Modelling with Oracle Designer

Last Updated: April, 2013

Version: 1.0.1

Document: NRS_Data Modelling Standards_Designer_1.0.1

Contents

Version Control

Version	Description of Change, Review or Approval	Author	Date
0.9	First draft of the guidelines consolidated from Natural Resource Sector data modelling standards • Guide S7 - Modelling Standards for Relational Applications (MFR) • Oracle Designer Standards V 2.1 (MOE/MAL) • Oracle Designer Standards V 2.0 (ILMB)	Tom Fulton	December 2011
1.0	Revisions to consolidated Data Modelling Standards guide after review with data architecture team members, other DA staff, other Architectures staff, Database Administration staff, the Standards Review Committee, Leadership Committee and IMB management.	Tom Fulton	
1.0.1	Change occurrences of Data Administration to Data Architecture. Update links to new NRS site	Bill Holland	April 2013
1.0.1	Changed title to NRS Data Modelling Standards with Oracle Designer to indicate that these are Designer standards and are different than NRS Data Modelling Standards with EA (Enterprise Architect).	Tom Fulton	July 2015

1 Introduction

This document was created by the Data Architecture / Data Management section of Information Management Branch as an initial consolidation and refinement of data modelling standards from the natural resource ministries.

Applicability of the appropriate type of data modelling – conceptual / domain, logical (relational / class), physical, reporting / warehouse – is determined during whiteboard sessions as outlined in section 1.4 - How to Apply Data Modelling Standards.

The data modelling standards described here reflect best practices from source ministries to create a single data modelling standard for the Natural Resource Sector.

From a business context, these concepts provide foundational support for defining and improving data and data sharing sector-wide, within government and to the public.

1.1 Purpose & Scope

Good data design is a critical building block for effective business data management, and for sector and government data sharing. Data modelling is the foundation of good data design.

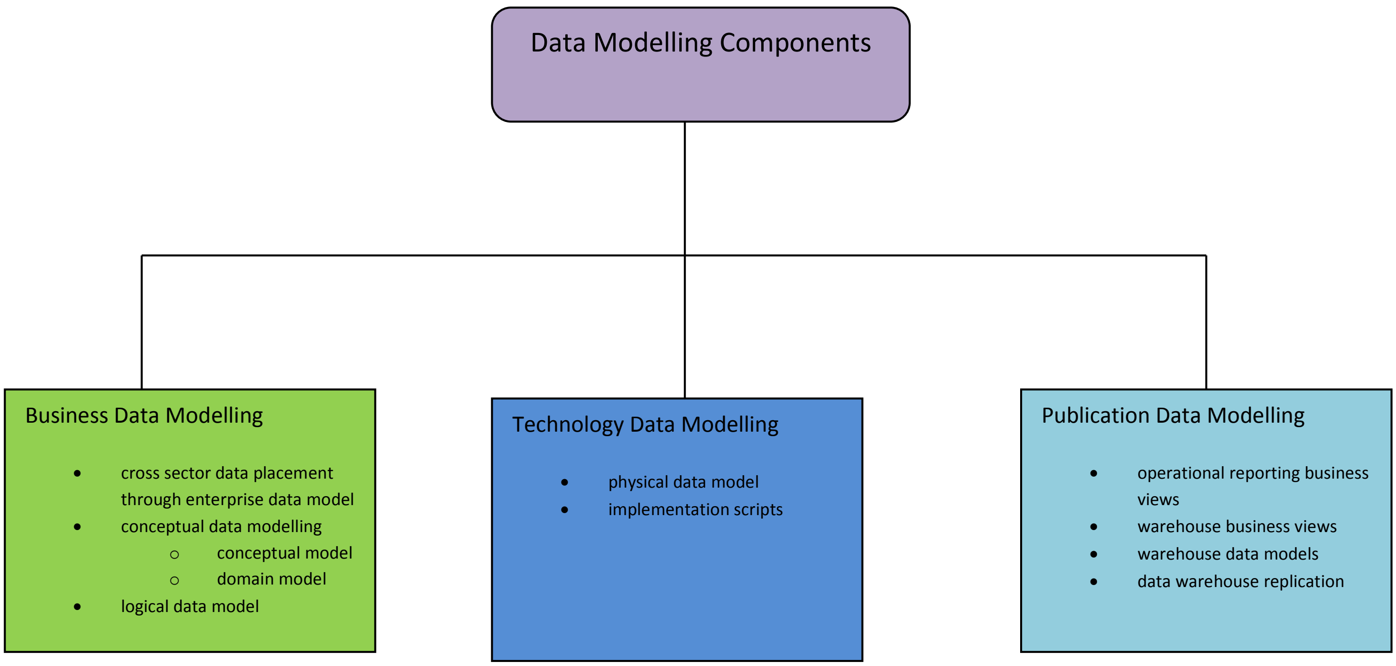
This guide illustrates and explains the form, standards, and conventions to be used for data models of applications intended for any of the information processing environments within the Natural Resource Sector.

It explains data modelling for the Natural Resource sector from a number of views:

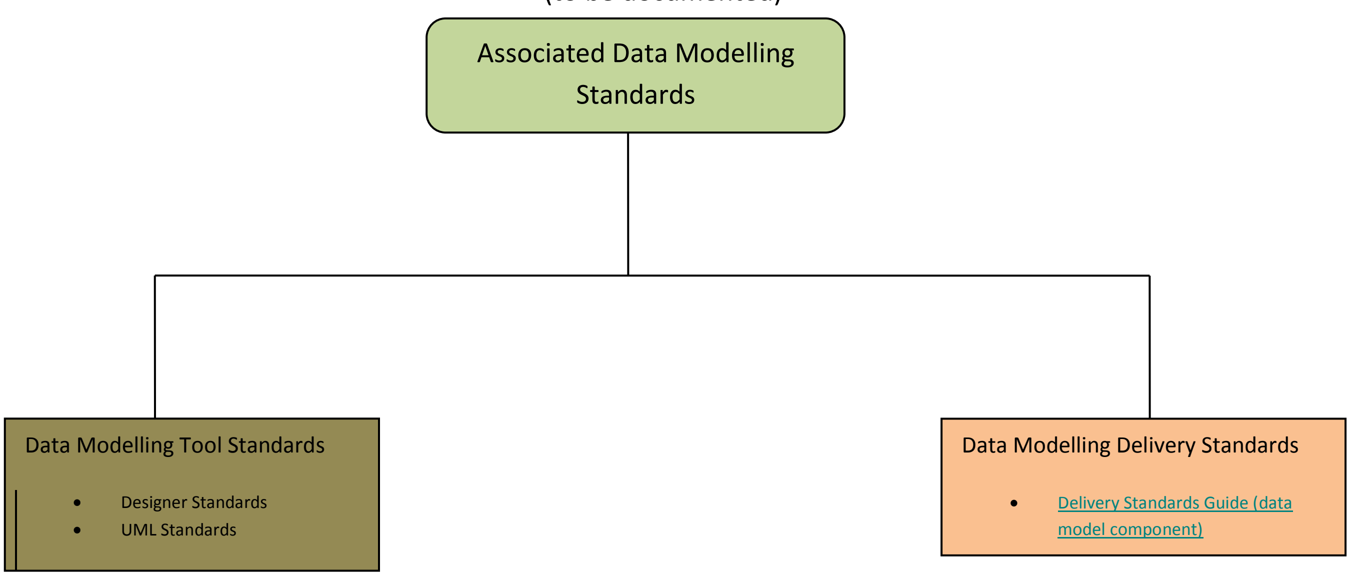
- from an enterprise data view for data architecture / data integration
- from a high-level business view for conceptual data design
- from a detailed business view for detailed data requirements definition and design
- from a technology view for physical data design and implementation
- from a publication view for data reporting and data warehousing

The guide is intended for data designers, data architects and data analysts who have some knowledge of the techniques and procedures involved in information engineering methodology. The data modelling standards that we employ are very rigorous, and we firmly believe they help tremendously in identifying and fixing problems earlier in the development cycle. The difficulty with rigorous standards is that those involved in the development process must first take the time to understand the standards, then must use them properly. If you have any questions about this guide or the data modelling process at for the Natural Resource Sector, see one of the [Data Architecture staff](#) in the Information Management Branch.

1.2 Data Modelling Components



(to be documented)



1.3 Related Standards and Documents

Name	Description
Relate to SDLC	
Relate to Business Requirements Specification	
Relate to Design Spec	
Relate to Data Quality	
Relate to metadata guide	
Relate to data modelling tools guide	
Relate to delivery guide	

1.4 How to Apply Data Modelling Standards

There are different types of application development across the sector requiring different types of data modelling. Operational, computational and publication applications are a few examples that may require different types of modelling (conceptual, logical, physical, warehouse) and different types of models (relational, object-relational).

The appropriate types of modelling and standards to be used in the development are determined at a technical White Board Session held at the beginning of the project.

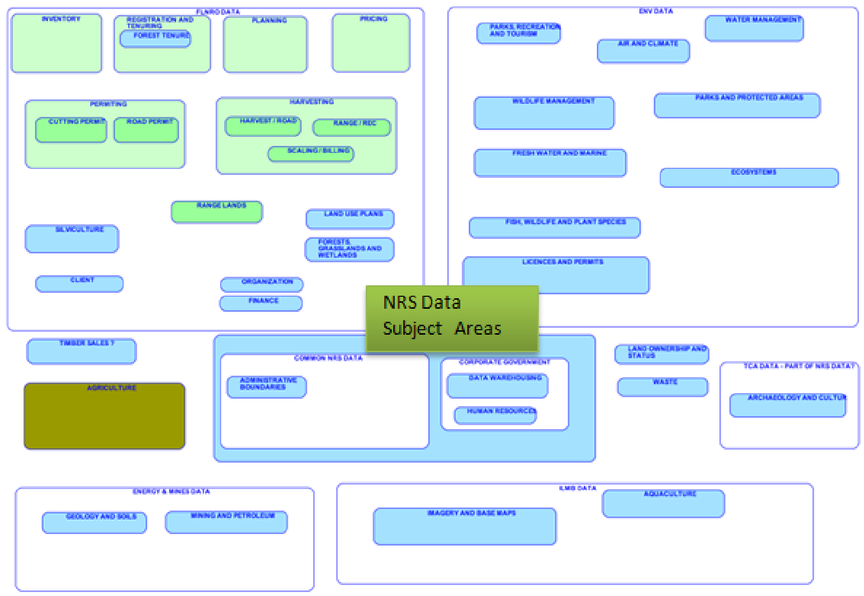
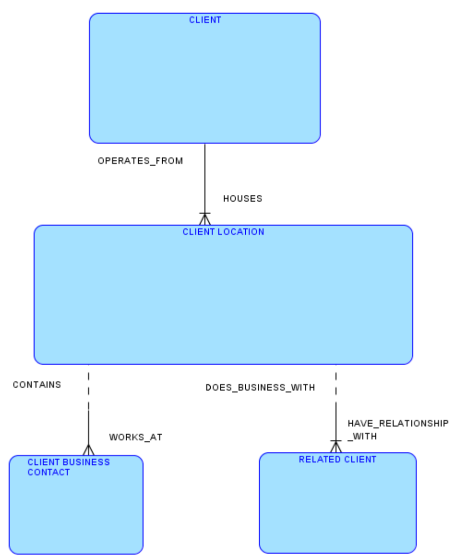
1.5 Items Audience Must Understand

Audience Member	Component of Guide
Business Portfolio Managers	BPMs, project leaders and staff from the involved business area(s) should understand the overall process for model validation
Data Designers (Data Modellers / Application Developers)	Must understand these parts of the guide thoroughly • high-level business view for conceptual / domain modelling • detailed business view for logical data modelling • technology view for physical and implementation modelling • publication / open data view for reporting and warehousing modelling • should be familiar with the concepts within the Enterprise View
Data Architects / Database Administrators	Will use these standards for project logical data model reviews and sector-wide data design
Database Administrators	Will use these standards for project physical data model reviews

Data Architecture / Data Management Services of Information Management Branch will ensure these standards are current, visible and used.

2 Data Modelling Views

There are a number of different types or views of data modelling within the Natural Resource Sector created and used for different purposes.

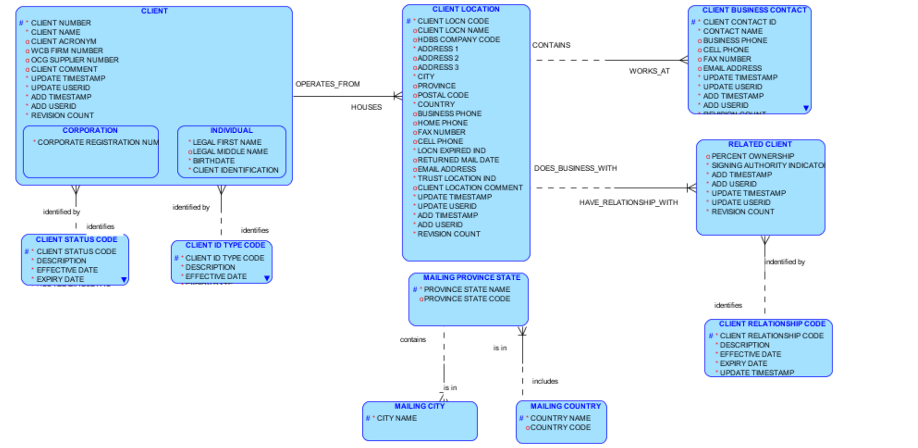
An enterprise view profiles data across the sector. High-level business views show general scope of the business data. Enterprise View (enterprise data model)	• subject area based profile of data across the sector • includes all major data subjects in sector • guidance for data placement, authoritative source, non-duplication and integration	
High Level Business View (conceptual data model, domain model)	• business area based profile of business data needs • includes all major entities and relationships • first step in organizing business area data requirements	

Detailed business views show the data defined as part of detailed business requirements. Technology views show physical data definition necessary to create database objects. Publication views show data designed for analysis and reporting.

Detailed Business View

(logical data model)

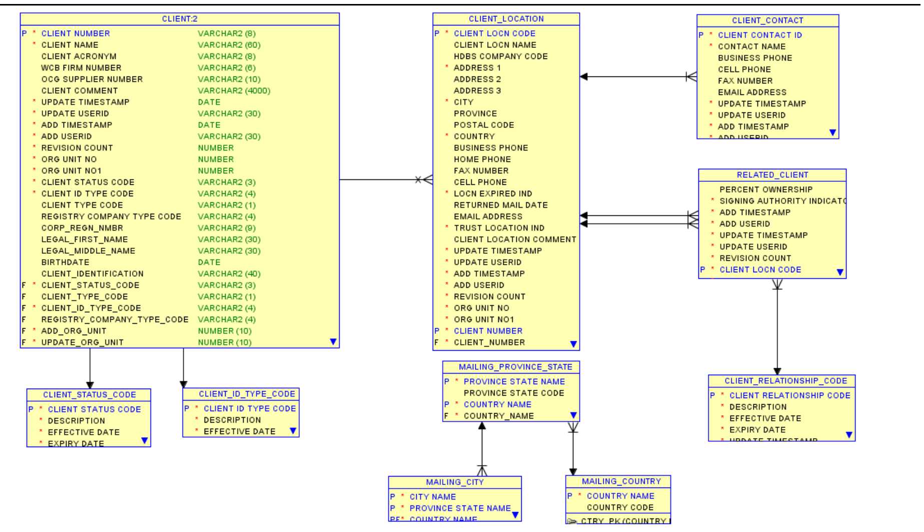
- business based detail of business data needs
- includes all business area entities, attributes and relationships
- detailed data requirements



Technology View

(physical data model)

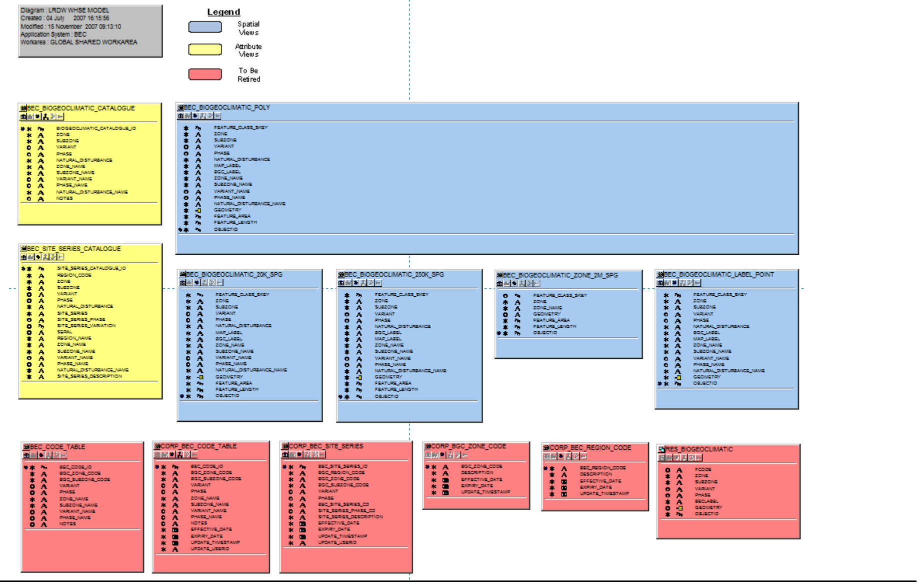
- technology based detail of business data needs
- includes all tables, columns, foreign keys, indexes, table generation script
- physical data design for database implementation



Publication / Open Data View

(warehouse / reporting data model)

- publication / reporting based view of business data
- includes business area physical data for publication
- denormalized / physical data design for reporting / warehouse implementation/ open data



2.1 Principles for Defining NRS Data

2.1.1 Why do we model our data?

<table>
<tr><th>We model our data:</th><th colspan="2">We model our data to get:</th></tr>
<tr><td rowspan="8">• to create data necessary to support business operations
• to ensure data is well defined by the responsible business area
• to understand where data fits within the organization
• to understand what data is available across the organization
• to ensure data is well understood by everyone using it
• to ensure quality data is available to support all business needs
• to help ensure data is well designed and managed as a critical business asset
• to avoid downstream impacts from changes to shared data
• to support greater BC government goals of open government and open data</td><td>The right data</td><td>The data I need</td></tr>
<tr><td>With the right completeness</td><td>All the data I need</td></tr>
<tr><td>In the right context</td><td>Whose meaning I know</td></tr>
<tr><td>With the right accuracy</td><td>I can trust and rely on it</td></tr>
<tr><td>In the right format</td><td>I can use it easily</td></tr>
<tr><td>At the right time</td><td>When I need it</td></tr>
<tr><td>At the right place</td><td>Where I need it</td></tr>
<tr><td>For the right purpose</td><td>I can accomplish our objectives</td></tr>
</table>

These are a few of the main reasons but it is by no means an exhaustive list.

We model our data to create source data that is the foundation for tracking and understanding Natural Resource business operations across the sector, within the BC Government and by the public.

2.1.2 Some basic modelling principles to help us achieve these goals

Single Source Data	**Defined once** - data is defined from a corporate sector-wide perspective with one business area responsible for defining the data and data standards to use throughout the sector **Captured once** - data is captured through one process and has only one source **Used many times** - data is shared throughout the sector, broader government and with the public
Non-Redundant Data	The sector does not define redundant data and has extensive attribute and spatial data holdings from significant investment so **determining if data exists** before defining new data is essential.
Quality Data	Defined so it is **fit for use** – for both business area use and use beyond the defining business area
Sharable Data	Data that can be is **open** without borders to its use and sector business areas plan and model for open data
Integrated Data	Where data has been historically defined within the sector by different business processes but has the same business purpose and use, multiple copies of the same data can be integrated to achieve **one process** for management of the data and **one source** for its availability.
Understandable Data	**Clear, intuitive** data definition allowing understanding of what the data is from its definition and description to allow the user to understand it's value to them

To benefit from data sharing and eliminate data duplication there are some search applications to help determine if data exists within the sector.

Integrated Data Dictionary	An inventory of operational and data warehouse Ministry of Forest application data.
Integrated Spatial Data Dictionary	An inventory of Ministry of Forest geographic data features.
Repository Object Browser	Oracle's browsing/searching tool for access to MOE/MAL data models.
BC Geographic Data Discovery Service	A discovery service with full data description and data quality information for Natural Resource Data and other government data held within the BC Geographic Data Warehouse.

3 Business Data Modelling

3.1 Objectives

The objectives of business data modelling are to:

- determine data placement within the sector data profile to identify responsibility, avoid data redundancy and promote data sharing
- define high-level and detailed data requirements necessary to meet business and sector needs

3.2 Enterprise Data View

3.2.1 Purpose

The purpose of enterprise data modelling is to create a subject area based horizontal view of data across the sector and to use this view as guidance for data placement, identification of data responsibility and authority (authoritative source), and for creation of integrated data eliminating overlap and duplication.

The enterprise data model is created and maintained by the Data Architecture section of Information Management Branch.

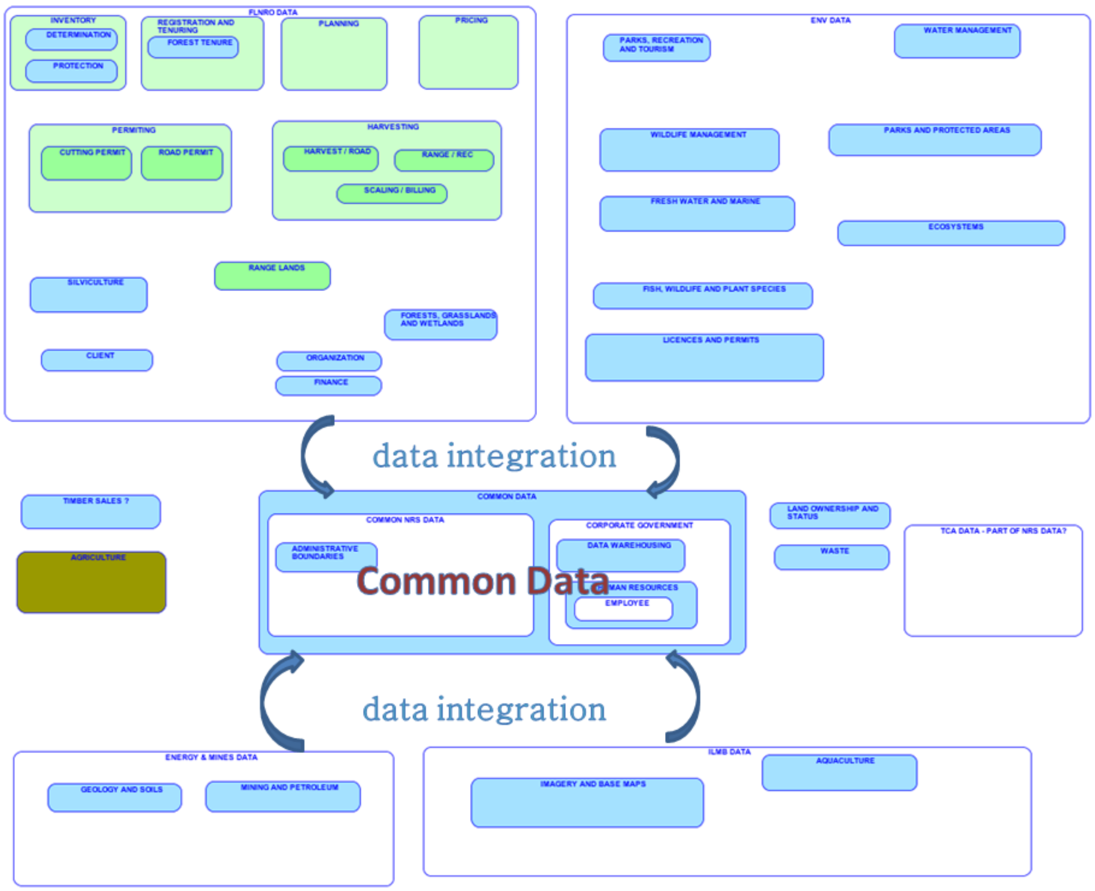
3.2.2 Context for Enterprise Data Modelling

Goals of the Natural Resource Sector	Goals of NRS data modelling
• **One Project, One Process** streamlines and coordinates all necessary authorizations for a single land based project in order to provide better service to clients and to improve the natural resources sector's internal administration efficiency	• one set of authoritative source data to support one process • sector data holdings designed to meet the business needs of the sector
• **Open data** are accessible, reliable, and useable by clients and businesses which meet their information needs for planning, investment decisions, transparency and engagement	• high quality data design and clear definition to support operational, data sharing and Open Data needs • appropriate spatial data design to support increasingly complex land use issues and natural resource decisions
• Reducing duplication for clients providing the same or similar information to various government workers, streamlining processes, finding efficiencies among business processes and engaging with clients in meaningful dialogue is driving change within the sector	• non-redundant data across the sector eliminating overlap and duplication • data integration within the natural resource sector where appropriate

• Core systems, which manage hundreds of millions of government revenue and critical land and resource data, need to be maintained while enhancements are undertaken	• support to maintain the substantial investment across the sector in critical land and resource data

3.2.3 Enterprise Data Modelling

3.2.3.1 NRS Enterprise Data Model



appropriate consolidation to integration using generalization

3.3 High Level Business View

3.3.1 Purpose

The purpose of the high-level business view is to create a conceptual view of data necessary to help determine the data needed to satisfy business requirements, determine the scope of data to be created, determine data placement within the sector data profile, and to determine data sharing opportunities to support the business need.

The conceptual data model is created and maintained by the Data Designer.

3.3.2 Conceptual Data Modelling

The conceptual data model shows the information needs of the business. It is designed at a conceptual level so shows the high-level data needs of the business and omits the detail.

There are two approaches currently used within the sector to create conceptual data models.

1. Relational data modelling uses relational modelling theory to define the major entities of interest to the business
2. Domain data modelling uses object-oriented principles to define the major classes within the domain of the business problem being addressed

3.3.2.1 Relational Conceptual Data Model

A conceptual ER model is a high level model defining the major entities within the scope of the business problem being addressed.

A conceptual ER model includes both a high-level entity relationship diagram and written descriptions of the entities. Conceptual ER models also show relationships between the entities being described.

3.3.2.2 Entity Relationship Diagram (Conceptual Level)

This is a graphical or pictorial representation of the entities and their relationships, defined by a high-level analysis of what information the business records and what business rules are used.

- The conceptual data model shows the information needs of the business. It is designed at a conceptual level identifying major entities
- Major attributes may also be included where they are obvious or important to the business at this stage
- May include some high-level sub-types of entities to identify different types of high-level business requirements

3.3.2.3 Entity Descriptions

Entities are described to sector NRS Data Naming Standards

3.3.2.4 Relationships

Relationships between entities are shown and must have meaningful names to identify the business data flowing between the entities. Where cardinality and optionality are available they should be shown.

3.3.2.4.1 Deliverables

Completed conceptual data model housed in NRS repository. Review material is created from this model.

3.3.2.5 Domain Data Model

A domain model is a high level model defining the major classes within the project domain

A Domain Model includes both graphical (diagrammatic) and written (textual) descriptions of objects within the problem domain or the software application. Domain Models also describe how the classes are structurally related to one another.

- The domain model shows the information needs of the business. It is designed at a conceptual level identifying major entities
- Major attributes may also be included where they are obvious or important to the business at this stage
- May include some high-level sub-types of entities to identify different types of high-level business requirements

3.3.2.5.1 Class Diagrams

Class Diagrams use classes and associations to describe the static structure of a system. Class diagram names are suffixed with "Diagram".

Classes represent abstractions of objects with common characteristics, and may be definitions of software classes rather than real-world concepts. In other words, they can model domain concepts or software classes.

Associations represent the structural relationships between classes and are named, showing multiplicities.

3.3.2.5.2 Class Specifications

Class Specifications, or Definitions, define and describe each class in a textual manner.

Class Specification Names are to be in UpperCamelCase with major attribution in lowerCamelCase. Class Specification Names are to follow Enterprise Naming Standards where applicable. Classes are to be stereotyped, if determined to have data persistence.

Class descriptions are described to sector NRS Data Naming Standards

3.3.2.5.3 Deliverables

- For relational data models the completed conceptual data model housed in NRS repository. Review material is created from this model.
- For domain data models the completed domain data model in XML Metadata Interchange (XMI) format. Review material is created from this model.

3.4 Detailed Business View

3.4.1 Purpose

The purpose of the detailed business view is to create a logical data model defining and documenting detailed data requirements as part of overall detailed business requirements. It can also be used to refine the scope of data to be created, determine data placement within the sector data profile where not previously determined, and to determine detailed data sharing plans.

The logical data model is created and maintained by the Data Designer. The conceptual model, if one exists, may be used as a starting point.

Note: Currently most logical data models within the sector are created using relational modelling theory although object-oriented principles to define class models are also used. Standards for the definition of class models will be defined in the next release of this standards document.

3.4.2 Logical Data Model Overview

Detailed logical data models are usually developed within the Natural Resource sector as part of an application development project. The purpose of the logical model is to show the data that the application must store in order to satisfy business requirements. It shows how this data is related and explores any integration requirements with business areas outside the scope of the development project. It is created without any specific computer environment in mind, so little optimization for performance, data storage, etc is done.

3.4.3 Logical Data Modelling

Logical modelling is an integral part of systems analysis and system design. It is platform-independent so there are no differences in required standards between the major ministry computer environments.

The logical data model consists of an *Entity Relationship diagram* (see Data Model Examples #1 and a *data dictionary* (see Data Model Examples #2 and #3). The ERD is used to show entities and display relationships between entities in a pictorial format. The data dictionary contains textual information about each entity.

Logical data models:

- identify all entities within the scope of the project and textually describe each;
- include the unique identifiers (primary keys) for each entity;
- include a complete list of attributes (data elements) for each entity and textually describe and define each attribute;

- show properly named relationships between entities, and note identifying relationships between entities where appropriate;
- are usually expressed in third normal form, although they need not be
- identification the ministry Data Custodian is also included during the logical data modelling process

Data Architecture staff review the logical model with the designer and ministry staff to ensure that integration requirements were considered and that proper data modelling techniques were followed. Developers should consult with Data Architecture staff as early as possible, to obtain any previously defined and relevant data definitions for use within their development projects.

3.4.3.1 Spatial Data Definition

Defining spatial data is very similar to defining attribute data in that it must be defined from a corporate business perspective. The same standards for data sharing and non-redundancy which apply to attribute data also apply to spatial data. Each spatial feature class (like an entity) is the responsibility of a single business area and is assigned a data custodian. The data custodian is responsible for defining the corporate data standards by which the ministry will capture and use the spatial feature data.

The logical side of defining spatial data is the feature class *business definition*. The definition includes:

- a meaningful, intuitive feature class name;
- a clear description of the spatial data item;
- identification of the type of spatial data (OGC Simple Feature Specification);
- identification of responsibility for the data standards (Data Custodian);
- identification of working level responsibility for the data standards (Data Standards Manager);
- identification of responsibility for the physical data management (Data Steward);
- identification of theme(s) which the feature class is used in

For example a Forest Licence Cut Block is a business object containing both spatial and attribute data. For the spatial data component of the Forest Licence Cut Block feature class is

- named: Forest Licence Cut Block
- described as: an area approved by the ministry for harvesting which must comply with a forest development plan
- defined as: a polygon feature
- Data Custodian identified as: Forest Tenures Branch
- Data Standards Manager: name of employee in Business Application Manager position
- Data Steward: none

- theme: Forest Development Plan

Spatial data is also shared among different NRS business areas, and with other ministries and industry business partners. Information on data dictionaries holding corporate data definitions is available in NRS Metadata Publishing (link to David's guide). Before defining new feature classes check corporate data dictionaries to ensure they do not already exist.

Information on Land and Resource data is also available through the Land Information BC Discovery Service.

3.4.4 Logical Data Model - Description

The logical data model consists of:

- Entity Relationship Diagram(s) (ERD)
- Entity and Attribute descriptions (the "Data Dictionary")
- Data classification using corporate or application level codes. See NRS Corporate Code Standards for sector code definition standards.

3.4.4.1 Entity Relationship Diagram (Logical Level)

This is a graphical or pictorial representation of the entities and their relationships, defined by an analysis of what information the business records and what business rules are used.

- The logical data model shows the information needs of the business. It is designed at the logical level (separated from the physical implementation) as much as possible, however, where a automated transformation process is used (logical to physical) some physical design considerations may be required within the logical model.
- Each entity is named and represented by its own rectangle. See NRS Data Naming Standards for data naming standards.
- Sub-types of entities can be defined where it makes sense to do so. However, DBA's may determine that there are performance impacts in how sub-types are implemented. See Oracle Designer documentation for a complete discussion on options for implementing sub-types in Oracle Designer.
- The types of attributes that are displayed on an entity are indicated by the symbol displayed to the left of the attribute name.

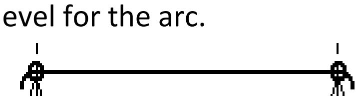
Symbol	Description
#	Unique Identifier (UID)
*	Mandatory Attribute
o	Optional Attribute

- Subset Entity Relationship Diagrams should be included where specific subject areas of the business should be shown

- A legend indicating the meaning of the various colours must be included on the diagram where colors are used to enhance readability

3.4.4.2 Relationships (between one or more entities)

- Relationships between entities are shown (see figures for notation used in Oracle Designer):
    - each relationship is drawn as a line connecting the two entities;
    - each relationship is given a name that indicates what information it imparts (relationships are named in **both** directions);
    - identifying relationships are indicated by a bar on the relationship line, which shows that a foreign key is being used as a primary identifier (or part of the primary identifier) in the entity closest to the bar and makes it easy to see which entities are dependent on other entities;
    - the *type* of relationship (*cardinality* and *optionality*) is specified as follows: the line style (dash or solid) indicates optionality and the relationship ends indicate cardinality. ***Relationship Types - Approximate Graphic Symbol - Oracle Designer(Barker Notation):***
        - Zero to Many Relationship (Left to Right) — — — — ———— < Many to One Relationship (Right to Left)
        - Zero to Many Relationship (Left to Right) — — — — — — — —< Many to One Relationship (Right to Left)
        - Zero to One Relationship (Left to Right) — — — — ———— One to One Relationship (Right to Left)
        - Zero to One Relationship (Left to Right) — — — — — — — — Zero to One Relationship (Right to Left)
        - Many to Many Relationship (Both Ways Zero to Many)

          >— — — — — — — —<
    - Note: 1:M relationships must be represented as 0:m on the ER logical data model. Any 1:M Mandatory requirements must be implemented through the application.
    - Mutually exclusive relationships are represented by an arc covering the 2 or more relationships included (see figure). Please note that the arc would need to be physically implemented by a stored procedure or trigger, since Designer does not generate anything at the physical level for the arc.
        - example diagram showing arc: 
    - recursive relationships can be used to implement hierarchies and are represented by a single relationship from an entity to itself
    - For domain/class data modelling notation refer to the [UML V2.4 Specification]

3.4.4.3 Unique Identifiers/Primary Keys

- Each entity (or entity occurrence) must be uniquely identified. This unique identifier or primary key (physical model) must be stable over time for the data record it identifies.
- The primary key may be made up of a combination of one or more attributes or relationships. It is preferable that this primary key is a stable business key.
- The data model should not contain an extra ID# (surrogate key) on each entity, but instead should use a proper BUSINESS KEY which is unique. If this is not available or the business key is volatile, then a surrogate key is okay, e.g. Project Id.
- If the primary key is a surrogate key then a unique business key should be defined to avoid duplicate records. Where the table is strictly a lookup table (data is pre-loaded not user entered) in some cases it will not contain a business key.

3.4.4.4 Entity Descriptions ("Data Dictionary")

The logical data dictionary textually describes the entities drawn on the ERD and includes the following information:

- Entity descriptions, which contain:
    - the name of the entity (from the ERD);
    - a brief textual definition of the use and purpose of the entity (see below - "*Definitions*");
    - names of attributes (data elements) appearing in the entity;
    - names that denote business aliases or physical naming decisions made at this stage;
        - Note: currently in NRS there are both singular and plural, and application acronym prefixed and non-prefixed approaches to physical table naming. The DA/DBA for the project will identify the correct pattern.
    - identifier(s) of the entity (the attribute(s) and/or relationship(s) that uniquely identifies an instance of that entity);
    - partition name(s) and corresponding subtypes, if the entity is further subdivided.
- Attribute (data element) descriptions, which document the following for each attribute:
    - name (see NRS Data Naming Standards)
    - definition (see below -- "Definitions");
    - data type (character defined as char, varchar2, numeric, date, etc.);
        - Note: currently in NRS there are both solely VARCHAR2 and CHAR and VARCHAR2 approaches to physical character data types. The DA/DBA for the project will identify the correct pattern.
    - data type of integer is not valid – must be number with specified length
    - length including variable length field considerations;
    - valid values or value ranges if necessary (see NRS Corporate Code Standards for sector code definition standards)

    - while the NRS does not define global domains for attribute formats, for some standard attributes such as ENTRY_USERID, ENTRY_TIMESTAMP application level domains may be defined

- Any coded values that will be used must be identified so that they can be reserved at the logical model review (i.e. coded values for lookups, or the use of standard ministry codes, for example silviculture technique codes, inspection type codes, etc.). See Classifying NRS Data Using Corporate Code Tables for details on code definition.
- Projects are allowed to create their own lookup tables to support table-driven rules e.g. rates, etc.
- "Large objects" (BLOBs, CLOBs) data type attributes may be separated into their own entities, normally with a 1:1 relationship to the main business entities. For geometry data type attributes, refer to the section High Volume Data Design for design considerations.
    - Note: currently in NRS some large objects are included in the business entity/table and some are separated into their own entity/table for physical data management. The DBA/ DA for the project will identify the correct pattern.
- Some projects will build temporary tables. If it is a temporary table to be used ongoing by an operational system, then include the table in the logical model. Don't include temporary tables created only for migration purposes.

*Foreign key (attributes) information is **not** textually recorded in the logical model*, since the existence of the foreign key is already defined by the relationship joining the two entities on the ERD.

Data Custodian is added as supplemental metadata later.

3.4.4.5 Definitions

Entity and attribute definitions must clearly describe what business information they record, using:

- *precision* - they resolve ambiguities and qualify imprecise terms;
- *completeness* - all terms are defined;
- *clarity* - **plain** English, few if any buzzwords;
- *brevity* - brief and to the point;
- *compatibility* - with other definitions.

Example - Good Definition	Example - Bad Definition
CLIENT - Any individual or registered company who is dealing, has dealt, or plans to deal with the ministry. This is effectively the client name and address list for the ministry.	CLIENT - A ministry client.

3.4.4.6 Attribute Order

To understand the logical model most easily, attributes should be ordered within each entity as follows:

- First: Primary identifier(s)
- Next: Attributes (these can be grouped by similar subject - e.g. address attributes are together)
- Last: entry and update userids or timestamps (unless they participate as a primary identifier)

Note: The above order will be required for the physical model column order.

3.4.4.7 Attribute Format Standards

The following specific format standards should be followed:

- DATE format for timestamp fields.
  - date fields should use format YYYY-MM-DD-HH24.M1.SS

3.4.4.8 Mandatory/Optional Attributes

- Primary Key attributes or relationships are always mandatory.
- Audit attributes are required for entities/tables which record transactions, or go through changes in state due to a business process and are recommended for other entities/tables. They are especially important for data warehouse replication to determine when data values have changed.
  - Note: currently in NRS there are different standards for audit columns. The DA/DBA for the project will identify the correct pattern. See NRS Data Naming Standards for naming standards.

3.4.4.9 Supporting Documentation (optional)

There may be other documentation that contains further information about the business area, or helps to describe the business in more detail. Supporting documentation may include:

- Statement of Scope (Terms of Reference)
- Requirements Document
- Process Model Diagram, Data Flow Diagram, or other useful diagrams

3.4.5 Logical Data Model Review Process

In general, attendees at logical modelling sessions must include: Data Custodian representative(s), Data Architecture representative, contracted staff responsible, and usually a DBA representative. The final approval meeting(s) should include a DBA representative for their information leading to the physical modelling effort.

There may be several modelling sessions before models to:

- gain an understanding of the model's business rules and relationships (the ERD)
- identify cross-program entities and any issues and Data Custodian resources to resolve issues if necessary
- review the technical quality of the model

A best practice is to work with the project DA early in the modelling process for information on existing, sharable data, to avoid creating redundant data, to understand NRS modelling approaches and standards.

3.4.5.1 Logical Model Checklist

At the logical modelling meetings, those present should look for:

- *Identification of cross-program entities* (i.e. entities of interest to another program area within the ministry) that may require other Data Custodian representatives to attend.
- *Identification of protection of privacy security risks* (i.e. access to data, e.g. personal information, that is private to the business) that may require an **access** userid and date/time be captured to verify proper data access.
- *Identification of historical data needs* (i.e. entities where data captured in the past has on-going relevance to the business) that may require capturing of historical data.
- *Technical quality of the Entity-Relationship Diagram*:
    - Is the ERD presentable with entities defined in a logical and readable manner?
    - Does the cardinality (one-to-one, one-to-many, many-to-many) and optionality (sometimes, always) of each relationship reflect the true business information requirement?
    - Were relationship names chosen properly? Are they meaningful, do they describe the business? (e.g. a relationship called "has" doesn't describe much)
    - Are identifying relationships marked properly?
    - If multiple relationship paths between the same entities are really required, then what are the reasons?
    - Are many-to-many relationships truly many-to-many? (90% are <u>not</u>)
    - Are mandatory one-to-one relationships really required or should the two entities be collapsed into one?
- *Technical quality of the Data Definitions*:
    - Do entity/attribute names clearly indicate the meaning of what they store? Where platforms restrict the length of column names, are the physical names identified? Do they follow the right physical model naming pattern (in Designer the entity Plural) for the environment the physical data will be created in?
        - Note: currently in NRS there are both singular and plural, and application acronym prefixed and non-prefixed approaches to physical table naming. The DA/DBA for the project will identify the correct pattern.

    - Do entity definitions accurately describe the business usage of what the entity contains (including any cross-program usage)?
    - Are entity identifiers (primary keys) properly defined, including identifying relationships? Are all identifier attributes mandatory (i.e. no blanks or nulls allowed in primary key columns)?
    - Are attributes listed in the proper order (primary identifier, attributes, update userids or timestamps, varchars)?
    - Do attribute definitions accurately describe the business usage of what each attribute contains? Is the datatype, length, valid value range(s) listed?
    - Do entities have the proper Data Custodian identified?
    - Do attributes fit the entity (i.e. proper normalization), none there that seem to belong somewhere else?
    - Do proposed physical names (abbreviations) follow NRS Data Naming Standards and are reasonable? Do the abbreviations make sense, capture meaning, don't intrude on other sector business areas? Are there no abbreviations unless necessary
        - Note: currently in NRS column lengths in some Oracle databases are restricted to 30 and in some others to 28. The DA/DBA for the project will identify the correct pattern.
    - Do attributes containing codes include a representative sampling of the code values? Are the codes listed new codes, or are they equivalent to or a subset of already existing codes? Does the information stored really need a code or would a Yes/No indicator be better used?
- *Code definition for the ministry's standard code tables*:
    - Is the code definition complete, i.e. description, custodian, update contact?
    - Is the list of code values defined? Refer to NRS Corporate Code Standards for standards for code creation.
    - Are the codes' characteristics (character type, length) reasonable, with thought given to how the codes' use may change in the future?

The data model reviews are usually conducted using the Data Modelling Review Template.(link)

3.4.6 Deliverables

- For relational data models the completed logical data model housed in NRS repository. Review material is created from this model. Where Designer is used a DMP file may be requested on earlier reviews.
- For object data models the completed class data model in XML Metadata Interchange (XMI) format. Review material is created from this model.

4 Technology Data Modelling

4.1 Objectives

The objectives of technology data modelling are to:

- define how business data will be physically designed for database implementation, data access and performance, and for storage management
- create the implementation scripts used to build the database objects

4.2 Database Design View

4.2.1 Purpose

The purpose of the database design business view is to create a physical data model necessary to define and document exactly how data will be physically organized in the database.

The physical data model is created and maintained by the Data Designer. It is the physical implementation of the logical model detailed business data requirements.

4.2.2 Physical Data Model Overview

The physical model is to shows how each data element will be implemented and stored on the database. Physical models may vary from the logical model in that the physical model may introduce objects that do not contribute to the business information requirements of the application. These new objects may be created in order to speed response times, ensure that the application fits within the physical limitations of the computing environment, improve maintenance turnaround, or for other performance or database tuning reasons.

4.2.3 Physical Data Modelling

For physical modelling, differences exist in the physical platform implementation, but standards for complete physical definition remain the same. For the ORACLE platform DA and DBA tools exist to implement the physical tables from the data definition.

Once the logical model has gone through the validation process, the designer generates a preliminary physical data definition using the logical data model as input (see Physical Data Model *data dictionary* (see Data Model Examples #4). The final version of the physical data definition will be a result of a joint effort between Database Administration (DBA) staff and the application developer. Data Architecture staff also participate to ensure information requirements identified in the logical model are translated properly to the physical model. DBA staff mainly ensures that the trade-offs between performance, data storage requirements, and application complexity are balanced appropriately.

Physical data models:

- are primarily concerned with physical limitations, performance and space requirements;
- need not be expressed in third normal form;
- identify all tables for the application;
- identify all columns listed in standard ministry column order;
- include definitions for all columns in each table;
- identify the primary index for each table;
- identify alternate indices for each table;
- show relating columns for each relationship (foreign keys).

De-normalization and the addition of columns to support special processing logic may be done as required. This must, however, be fully documented in the description for the column and must be done with DBA approval. If there are certain requirement enhancements to major objects in the physical design that are not performed at the logical level the changes done to the physical implementation that are not reflected in the logical design must be fully described.

4.2.3.1 Spatial Data Definition

The physical side of defining spatial data is the platform implementation definition. For MoFR operational spatial data this definition will mostly be in Oracle Spatial. For MOE/MAL/ILMB/MEM operational spatial data is usually defined in ESRI ArcSDE.

<u>High Volume Data Design</u>

For physical data management, some high volume data may be stored in its own table. CLOBS and BLOBS will **often** be put in a separate table related to the business table. Geometry, depending on the density of the feature, **may** need to be put in a separate table. If the feature is dense, DBA may require the geometry to be defined in a separate related table - as high volume data could cause physical data management problems such as lengthy unloading and loading times for data changes. The DBA/ DA for the project will identify the correct pattern.

4.2.3.1.1 Oracle Spatial Layer Definition

Oracle Spatial is used for some sector operation spatial data. For Oracle Spatial Layers the layer is created through an Oracle's Spatial Data Option (SDO) geometry attribute:

- a GEOMETRY attribute on the business entity,
- a standard attribute name, data type and length for the geometry attribute - GEOMETRY type SDO GEOMETRY;
- a clear description of the spatial data item - put in the GEOMETRY attribute description

4.2.3.1.2 SDE Layer Definition

SDE is also used for some sector operation spatial data and for spatial data being replicated to BCGW. For SDE Layers created for the BCGW the layer is created through a geometry column on the table. This includes the business definition information above plus:

- a GEOMETRY attribute on the entity with the spatial component which will be used to create the layer;
- a standard attribute name, data type and length for the geometry attribute - GEOMETRY Number 38;
- a clear description of the spatial data item - put in the GEOMETRY attribute description

4.2.4 Physical Data Model - Description

The physical data model consists of a physical model diagram, supporting table and column definitions and Data Definition Language (DDL).

4.2.4.1 Oracle Physical Model

4.2.4.2 General Notes for the Physical Data Model

The following items should be noted:

- The table definitions show the physical structure of the records (tables), fields (column definitions), and indexes which will be implemented in the database (for an example see Data Model Examples, Example 5).
- There must be a one-to-one match between the table definitions and the database tables being implemented on the physical platform.
- Logical model subtypes cannot exist in the physical model; they are resolved into one or more tables during table generation. The business data structure is identified in the logical model and can be referred to there if necessary.
- Where a surrogate key is used as the primary key standard practice is to also define a unique business key to avoid duplicate records and where this practice cannot be met approval from DBA must be obtained.
- The order of columns (fields) for the physical tables is finalized to ministry standard column order in the table definition.
- Indexes needed for physical implementation are finalized in the table definition.
- In Oracle, if the primary key is a single surrogate key, an Oracle sequence can be used to generate the value for the id. However, if the primary key is a concatenated key, made up of a business key + a surrogate key (sequence), sequencing within the business key is not supported automatically by an Oracle Sequence. Therefore, if this is required, the application must build a separate routine to select max (surrogate key) for a given business key and increment by 1.
- DBA’s prefer stored procedures for implementation of complex business rules.

- Use stored procedures or triggers instead of specifying default values for columns.
- Provide physical table size & growth estimates (# of rows, expected yearly growth). Describe in documentation, not in DDL.
- Additional DDL standards, additional tips for developers are found on the Database Administration web page.
- Both DA and DBA's review DDL.
- For a description of the application delivery process, including DDL, please see the NRS Application Delivery Standards.

4.2.4.3 Column Order

For best physical database design, columns should be ordered within each table as follows:

- First: Primary identifier(s)
- Next: Attributes (these can be grouped by similar subject - e.g. address attributes are together)
- Last: entry and update userids or timestamps (unless they participate as a primary identifier)

DBA may require BLOBs and/or CLOBs to be placed in a separate table.

The DBA staff are aware of these considerations and take them into account during physical model reviews.

4.2.4.4 Application Code Tables

Codes in the sector are often managed corporately for maximum sharing and minimum redundancy. These codes are physically implemented within Application Code Tables for code lookup and referential integrity. The physical structure of an Application Code Table is created through the data modelling process. In some cases a replication process is used to automatically populate and update Application Code Table values from a corporately managed code source. In other cases load scripts are used. The DBA/ DA for the project will identify the correct pattern.

Please see NRS Corporate Code Standards for more information about Application Code Tables.

4.2.4.5 Referential Integrity

A relationship between entities will result in the generation of a foreign key constraint in the physical table definition. The resulting data column(s), if populated, must have a corresponding entry in the associated table. This is referential integrity. Also, entries in the associated table, e.g. lookup table, cannot be removed if there are data columns referencing them.

Referential integrity (Foreign Key constraints) is generally left on. The only exceptions are if the target table is sourced from an external database, or to temporarily remove foreign key constraints for conversion purposes.

Cross Server RI cannot be implemented through foreign key constraints. However, it can be implemented through triggers, but there would be a performance hit. Therefore, generally it is not recommended.

4.2.4.6 Supporting Documentation

Examples:

There may be other documents that show additional information about the business area, oriented towards physical design decisions.

Column naming for foreign key columns should be the same as the parent table column but sometimes there if there is justification the foreign key column name may be different. For example you might have a geographic and an administrative org unit in the same table which are both foreign keys to the same ORG_UNIT_NO column.

4.2.4.7 Managing Views - Data Access in an Integrated Environment

Data integration in systems development raises some issues regarding physical access procedures within the integrated database. The following is a brief outline of how general table/view access and security is currently managed.

- The use of interactive SQL in the production environment is strictly controlled.
- The Data Custodian of the table is responsible for ensuring that the ministry's information needs are met by that table. The Data Custodian is responsible for authorizing access to a given table. If another client requires the use of only some of the columns of that table, and is limited from using the full table, then that client (i.e. the "external user") must build a view, have it approved by the Data Custodian, and implement it so that all required ministry areas can access the appropriate data.
- Multiple views can be used to manage access to sensitive fields. The initial view built includes all columns on the base table. A subsequent view can limit access to only non-sensitive columns. "External" systems will likely use the non-sensitive views (*external* here means those systems not developed by the Data Custodian of the table). However, the number of views defined should be kept to a minimum, for example only in exceptional circumstances would there be more than two views defined for any particular table. see NRS Data Naming Standards
- There must be no views with "select *" as part of the view definition.

4.2.5 Physical Data Model Review Process

Data architecture staff participate in the physical data model review process to ensure the physical model is translated properly from the logical data model.

4.2.5.1 Physical Model Checklist

The following checklist is from the Data Architecture perspective only. At the physical modelling meetings, those present will:

- Review logical entities and corresponding physical tables.
- Review physical model printouts with the logical data definition:
    - Do proposed physical names (abbreviations) follow *Guide S19* abbreviation standards, and are reasonable? Do the abbreviations make sense, capture meaning, don't intrude on other ministry business areas? Are there no abbreviations unless necessary?
    - Are the logical entity identifiers appropriately and properly translated into physical keys in the physical tables?
    - Are all relationships on the logical ERD properly translated into foreign keys on the appropriate table, using the name of the primary key(s) as defined in the foreign table?
- Review any manual changes made between the logical and physical models. These should be described in a separate document. Are these manual modifications really necessary or could they have been avoided?

4.2.5.2 Database Administration Checklist

Database Administration (DBA) staff review the model looking at physical design decisions, including those for database efficiency, and may request further changes to the physical model. In general, DBA staff review:

- Index requirements
- Datatype selections
- Column order
- Review physical model:
    - Are entry points (indexes) fully defined ? Do they support all database access requirements?
    - Are space estimates available for all tables?
    - Are specific views of the data necessary for performance or security reasons?

DBA will not action any test releases until any associated table DDL (eg. Create table scripts) have been approved by Data Admin.

See DBA staff for more complete descriptions of their physical data model review concerns.

4.2.6 Deliverables

- For relational data models the completed physical data model housed in NRS repository. Review material is created from this model.

4.3 Database Implementation View

The database implementation view consists of scripts that can be run in the database to create the physical objects designed in the physical data model. These scripts, Data Definition Language (DDL), are normally generated from the physical data model

4.3.1 Oracle Table Generation

In Oracle, a physical data model containing the implementation tables and columns is generated from the logical data model. Logical model to physical implementation mappings are created and refined during the generation process. As the physical model shows the table, column (in ministry standard column order) and foreign key structures that are the basis of all relational databases, once implementation mapping has been refined and the physical model generated there should be very little tweaking. Any modifications should be documented and a list submitted with the physical data model.

4.3.2 Oracle Table Generation Scripts (DDL)

Table creation scripts, or DDL scripts for Oracle databases are generated from the physical data model/table definitions in the Oracle Designer Repository.

4.3.3 Deliverables

Completed physical data model housed in NRS repository. Review material is created from this model.

5 Publication Data Modelling

5.1 Objectives

The objectives of publication data modelling are to:

- determine appropriate data design for reporting and data warehousing
- determine appropriate design for population / replication to reporting / warehousing databases
- determine publication needs to support Open Data

5.2 Operational Reporting View Modelling

5.2.1 Purpose

The purpose of the operational reporting business view is to create database view(s) of the operational business data optimized for reporting purposes.

Reporting business views are created and maintained by the Data Designer, either an in-house or a consultant data modeller.

5.2.2 Reporting Model

a) An application is considered "Reporting" only when using 100% existing data.
If the data is in its original structure or using a subset, no model is required.

b) If there is a changed structure (e.g. denormalized tables, concatenated columns, data summarization--this is often called a data warehouse), a model is required.

The future of "reporting" is direct web access to a standard set of corporate (replicated) databases, or a dimensionally-modelled data warehouse(s).

5.2.3 Data Currency

Users must be aware of timing mismatches when relating data which may have been transferred to the reporting database at different times, and may therefore have different levels of currency and accuracy.

5.2.4 Deliverables

Completed operational reporting data model views housed in NRS repository. Review material is created from this model.

5.3 Warehouse Business View Modelling

5.3.1 Purpose

The purpose of the warehouse business view is to create a physical data model and/or physical data view(s) defining and documenting the tables and views that will be implemented in the data warehouse.   It also defines and tables and view used in replicating data to the warehouse from the operational system.

The warehouse data model is created and maintained by the Data Designer, either an in-house or a consultant data modeller.

5.3.2 Warehouse Data Model

There are two approaches currently used within the sector to create data warehouse models.

1. A data warehouse model, detailing the physical data to be stored in the data warehouse, is created as a separate model from the operational data model.
2. A business view, detailing the physical data to be stored in the data warehouse, is created within the operational data model.

5.3.2.1 Warehouse Data Model Approach

Currently NRS data from the MOE/MAL ministries, which will be replicated to the BCGW, is defined in a separate Oracle Designer data model. The data is defined as materialized views. These materialized views are created to adhere to BCGW standards and for streamlined effective business views.

5.3.2.1.1 Deliverables

Completed warehouse data model housed in NRS repository. Review material is created from this model.

5.3.2.2 Warehouse Business Views Approach

Currently NRS data from MoFR, which will be replicated to the BCGW is defined as, often denormalized, business views within the operational data model see Data Model Examples #4 (Physical Data Model) and *data dictionary* (see Data Model Examples #5). Three components are usually defined:

1. Table(s) which is implemented on the operational side and used to accumulate the data to be replicated to the warehouse. These tables are normally used to select and sometimes transform data from multiple operational tables and often use complex queries to select and transform the data.
    - it is required that the extract query be documented in the notes section of the table used for data accumulation.

2. Corresponding table(s) which are implemented on the warehouse side and used to store the data replicated to the warehouse. These tables are normally the same as the operational data accumulation table so are defined only once in the model but implemented on both the operational and warehouse environments.
3. A user warehouse data view, often spatially enabled, of data in the replicated table which removes columns used for replication tracking and management but are of no interest to the user.

These tables and views to be implemented in the BCGW follow the standards for BCGW warehouse data design.

5.3.2.2.1 Deliverables

Completed warehouse data model business views housed in the operational data model within an NRS repository. Review material is created from this model.

5.3.3 Data Quality

To understand the fitness for use of data replicated to the BCGW metadata must be added to the Geographic Data Discovery Service describing the data.

5.4 Open Data Considerations

When defining data to be replicated to the BCGW, the opportunity to include the data in the Data BC Open Data catalogue must be considered.

6 Data Model Examples

The following examples have been taken from various applications, all of which have been developed at different times.

The examples have been labelled to indicate how each was produced in Oracle Designer or other tool.

- Example 1: Logical Data Model (Oracle E-R Diagram) - a subset of the FTA Logical Data Model.
- Example 2: Sample Supporting Documentation (Oracle Entities and Attributes Report) - for the FTA Logical Data Model.
- Example 3: Sample Supporting Documentation (Oracle Tables and Columns Report) - for the FTA Physical Data Model.
- Example 4: BCGW Warehouse Data Model (Oracle Server Model Diagram) – for the BCGW subset of the FTA Physical Data Model.
- Example 5: BCGW Warehouse Oracle Table Definition (Oracle Tables and Columns Report) - – for the BCGW subset of the FTA Physical Data Model