

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

This document provides a review of the data model for <INSERT PROJECT NAME>.

Category of Review	Results of Review
A. Data Model Status	
1. First Cut Data Model? <ul style="list-style-type: none"> • Have the client area had a detailed walkthrough of the data entities, relationships and list of attributes? • Have optionalities, cardinalities been confirmed? • Have modifications been completed as a result of walkthroughs? 	History is as follows: <ul style="list-style-type: none"> ➤ 1st cut model provided – ➤ Initial Walkthrough meeting held – 2nd version provided with Revisions– ➤ Results of Review meeting –
2. Verification Process leading to a validated Data Model. <ul style="list-style-type: none"> • 1st Cut complete? • Has Data Admin review been completed? • Have modifications been completed as a result. 	<ul style="list-style-type: none"> ➤ Final version provided ➤ Verified ➤ DDL provided ➤ Verified ➤ All interdependencies resolved?
B. Data Model Scope	
1. Entity List Complete? Does the scope of the entities appear to match the project scope?	
2. Are temporary tables reflected in data model?	
3. Are all attributes correctly represented?	
4. Are all relationships correctly represented?	
4a. Cardinalities correct? Optionalities correct?	
4b. Relationship names?	
4c. Identifying relationships?	
4d. Multiple relationships between two entities?	
4e. Many to Many's truly needed?	
4f. One-toOne's truly needed?	
5. Are all primary keys correctly	

<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>

Category of Review	Results of Review
represented, mandatory. Relationships are included in primary key definition when needed?	
6. Foreign keys are not duplicated as attributes?	
7. Attribute Order (primary id, attributes, update userids or timestamps).	
8. 1:M relationships must be represented as 0:m on the data model. 1:M Mandatory requirement must be handled through the application.	
C. Data Definitions	
1. Entity Definitions – complete?	
2. Attribute Definitions – complete?	
3. Attribute Formats <ul style="list-style-type: none"> ➤ Varchar2's only, not chars ➤ Numbers not integers ➤ number with decimal, ➤ date ➤ timestamp 	
4. Attributes Mandatory/Optionality confirmed? –	
D. Naming	
Entity Naming	
1. Entity Names spelled out and correct for scope of entity	
2. Entity names should not be prefixed by the project name, except in some situations.	
3. Application Code table naming – confirm proper naming, use of subsets.	
Attribute Naming	

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

Category of Review	Results of Review
4. Primary Keys for Data tables should closely match entity names.	
5. Primary Keys for Application code tables must match entity names.	
6. Other attributes should fully spell out words unless affecting other apps already using table. Abbreviate only if over 30 chars.	
E. Design	
1. Historical Data Needs - Is the project to keep history? Are there conversion requirements?	
2. Projects can use their own discretion in whether or not they want to use sub-types or supertypes, as long as it makes sense.	
3. Data model should not contain extra ID# (SKEY) on each entity, but instead should use proper business key which is unique. If this is not available, then a surrogate key is okay.	
4. Check usage of UPDATE _timestamp and other date columns.	
5. Use of sequences for Sequence numbers?	
6. Referential integrity – This is strongly recommended. Only exceptions would include tables still sourced from the other platforms, or other exceptions may be made on a case by case basis.	
7. Normalization and Model Design, Design	The following questions are provided by Diagram to confirm the data requirements and

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

Category of Review	Results of Review
Questions/Issues and Revisions	solicit discussion: ➤
List subject areas or diagram names here and list any questions for each.	
8. Domains should not be used/documentated in Designer, since then they would need to be maintained there. Data Model can show examples only.	
F. Entity Sharing/Re-use	
1. Are all Cross-program entity types identified? e.g. Client, Org Unit, Tenure entities, etc. (Use shading to distinguish these. See section H. Model Format).	
2. Projects need to use application code tables for referential integrity and code validations. Confirm that existing Oracle code tables are used. Subsets should be created of existing code sets to isolate the application from changes to the master list.	
3. Confirm that new Application Code tables proposed are properly defined and are not redundant within the project. Also confirm there are no embedded hierarchies within the codes. Again, use subsets.	
4. Projects must submit a spreadsheet containing their proposed application code	

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

Category of Review	Results of Review
tables and new values required. (Format available). This is reviewed and verified against the data model.	
<p>5. Codes:</p> <ul style="list-style-type: none"> ➤ Naming correct (not too generic) ➤ Completeness ➤ Custodian ➤ List of Values ➤ Char Type ➤ Length - When deciding on length of code key, the length should be that of the longest value. 	
<p>6. Interdependencies with other Applications</p> <ul style="list-style-type: none"> ➤ Is this project changing any tables, that are currently used by other applications? ➤ Have all clients of existing tables to be changed been notified of the changes? ➤ Have all interdependencies been resolved in data model, and decision made as to which project will change their model, and/or maintenance release scheduled? 	
G. Privacy	
<p>1. Privacy Issues/Security Risks – Are there any?</p>	
H. Model Format	
<p>1. Entities which are going to be re-used from existing entities/tables should be color coded to show the groupings by Subject Area, e.g. client, tenure, and also code tables in general (sourced from code</p>	

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

Category of Review	Results of Review
list/subset), main entities for project.	
I. Preparation for Physical Model and Table Generation	
1. Follows Physical Naming Stds (S19).	
2. Modifications to Physical Model after generation are minimized and documented.	
3. Do not use option to set column prefix.	
4. Volume information should be prepared in preparation to give to DBA's. Could be entered in Designer.	
J. DDL - is reviewed by DA to ensure it matches the data model.	
1. DDL should be submitted by project accompanied by a list of all entities in scope, and associated tables to be created by this project or shared from existing tables. (e.g. what tables do they expect to be in TEST already). Sample format is available for their use.	
2. Comparison to Data Model entity list. a) Entity list matches data model. Not generating shared common tables?	
b) Primary Keys and Foreign Keys defined as per data model?	
c) Attributes naming and formats look okay?	
3. Application code tables: a) DDL should not contain scripts for inserting values into code tables.	

**<INSERT PROJECT NAME> Data Model Review
V1.3 – <DATE>**

Category of Review	Results of Review
b) Has their application code table spreadsheet been approved by the users? (It should be approved by users before DDL is approved.)	
4. Validation tables – Projects should submit scripts for loading their other validation tables/lookups that are not going to be in code list/code subset.	
5. Other things to check: <ul style="list-style-type: none"> ➤ synonyms ➤ defaults ➤ arcs implemented for mutually exclusive relationships ➤ check constraints – only used on exception basis ➤ views ➤ sequences ➤ don't include drop table commands ➤ - don't use alter scripts 	
6. DDL should not create tables that are already in Test or Production unless this change is included in the scope of the project.	
K. Data Custodianship	
1. Has a data custodian been identified for each entity?	
L. Records Management	
1. ARCS/ORCS – to be developed?	
2. Has official record been designated?	
3. Retention schedules – Has retention of electronic records been discussed?	