A Physical Database Model
for the
State of Iowa
Department of Transportation's
Linear Referencing System
TECHNICAL REFERENCE DOCUMENT

June 2000

Prepared by

GeoAnalytics™

1716 Fordem Avenue
Madison, WI 53704
Phone: 608-241-7100
Facsimile: 608-241-7116
E-mail: office@geoanalytics.com
URL: **http://www.geoanalytics.com**

TRANSDECISIONS

11835 W. Olympic Blvd.
Suite 700E
Los Angeles CA 90064
Phone: (865) 988-8345
Facsimile: (865) 988-8744
E-mail: **JulianRay@TransDecisions.com**

With

Oracle Corporation

222 West Las Colinas Blvd.
Consulting
Suite 1000
Irving, TX 75039
Phone: (972) 409-3907
Facsimile: (972) 409-3041
E-Mail: jlwinter@us.oracle.com

Iowa Department of Transportation

800 Lincoln Way
Ames, IA 50010
Phone: 515-233-7770
Facsimile: (515) 239-1828
E-mail: wschuma@max.state.ia.us

Table of Contents

Table of Figures

A Physical Database Design Model
for the
Iowa Department of Transportation
Linear Referencing System
Technical Reference Document

1 OVERVIEW

The Physical Design is the third and final design phase in the Iowa Department of Transportation (DOT) Linear Referencing System (LRS) Development Project. The purpose of the Physical Design is to determine how to meet the LRS requirements from the Logical Design phase using specific technology choices. One product of the Physical Design is the database model. This model describes the LRS database that was designed to meet the LRS data requirements. The LRS database will be in Oracle and contain Oracle Spatial and Oracle Spatial LRSx objects.

This document describes the physical database model. The document is referenced by the Physical Design Technical, Physical Design Summary, and Decisions Behind the Physical Design documents. There are two key model components provided here: a physical database diagram and a data dictionary. The physical database diagram illustrates the specific data structures and relationships that will be needed for the LRS. This document introduces and presents the graphical diagram and its contents.

In addition to the graphical representations of database elements, the physical database model contains table description information called a data dictionary. The data dictionary is stored as text fields associated with the diagram shapes. The data dictionary can be output in several formats. One output is a report of all entities and their relationships. Another output is a Data Definition Language script that can be used to directly create the design structures as actual objects in the Oracle database. The entity report is included at the end of this document.

2 MODEL DIAGRAM

2.1 Background

A physical database model diagram is used to explicitly define structures and relationships of requirements identified by the logical design. The general needs and system components described in the logical design need to be translated into a real world application environment. Due to the actual features or limitations associated with the particular software and hardware solutions chosen, database entities can be split or recombined. These new configurations must still continue to meet the requirements outlined in the logical design.

There are also limitations encountered when selecting the modeling software. The project participants selected Visio Enterprise 2000 for this project based on its ease of use, affordability and familiarity. The software has limitations in supporting custom database objects required by the LRS, but the Physical Design Team devised solutions for these limitations.

2.2 Graphic Components

Visio incorporates two intelligent shapes that form the foundation of the diagram. The first is the "Entity" shape that is used to define table elements and their internal column structure (Figure 1). Most tables contain the features shown in this figure. At the top is a table name, followed by the table primary key column. The rest of the rows are additional table columns. All of the column names are accompanied by the Oracle datatype of the column. Some column names are followed by the notation "(FK)", which indicating that the column is a foreign key referencing a column in another table.

Figure 1: Diagram Entity Shape

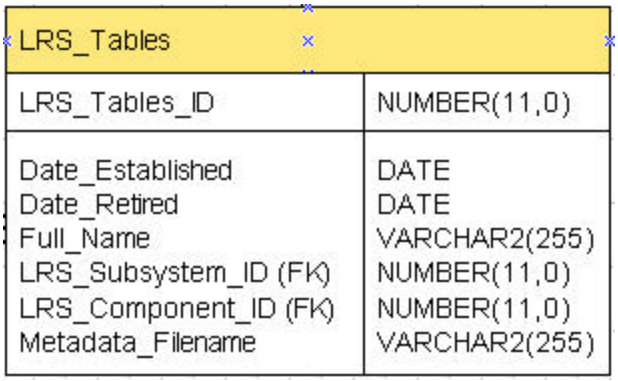
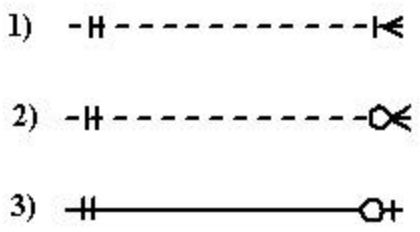


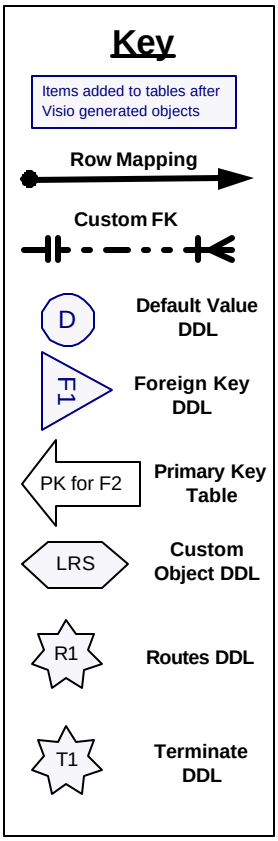
Table and column names in the diagram are displayed as mixed upper and lower case labels. This is for readability only; the data creation scripts generated from the diagram produce tables with all upper case names. Please refer to the appendix on data standards for more information (The second intelligent shape is the "Relationship" connector that connects two table entities (Figure 2). The connector represents a primary key / foreign key relationship between two tables (parent / child). This notation, referred to as IDEF1X in Visio, indicates that there is an explicit correspondence between the value in a column in one table and the value in a column in another table. This correspondence is enforced by the database and can be "one to one or more" as in connector #1, "one to zero or more" as in #2, or "one to zero or one" as in #3.

Figure 2: Diagram Relationship Connectors



The last set of graphic components is basic Visio drawing shapes that have been assigned custom property attributes (Figure 3). These shapes are used as work-arounds or enhancements to the database description shapes that are built into Visio. The intent was to describe as much of the database properties as possible within the graphical interface. These shapes are then used to generate the full set of database creation scripts without additional editing. Some of these shapes represent custom datatypes in Oracle, some indicate programmatic derivations of column vales, while some call a set of PL / SQL scripts with arguments to replicate structures across a number of database schemas.

Figure 3: Shapes With Custom Properties



- Custom shapes that generate alter table DDL code to add columns of extended object datatypes (SDO_GEOMETRY, LRSX_CONFLATION)
- Indicates a programmatic mapping of conflation objects to a portion of a geometric feature
- Indicates a custom process that maintains the correspondence between key values in two tables.
- Custom shapes that generate alter table DDL code to define default values for table columns
- Custom shapes that generate alter table DDL code to define foreign keys for widely separated tables
- Indicates the table to which the foreign keys defined above point to.
- Custom shapes that generate extended Oracle objects, Oracle schema, and overall DDL flow management scripts.
- Custom shapes that generate DDL calls to PL/SQL scripts that replicate route tables to route schema.
- Custom shapes that generate DDL code to cleanup after database table creation and end the process.

2.3 Diagrams

The following section contains the physical database diagram. The diagram has been broken into a series of letter-sized sheets that consist of related groups of tables. The relationship connectors extend across sheets in some cases. The adjoining sheet can be found by referring to the sheet map (Figure 4). An E-size plot can be made available from the Iowa DOT Project Manager, Bill Schuman.

Figure 4: Diagram Sheet Map

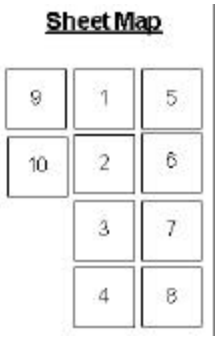


Figure 5: Diagram – Datum Subsystem Primary Tables (Sheet 1)

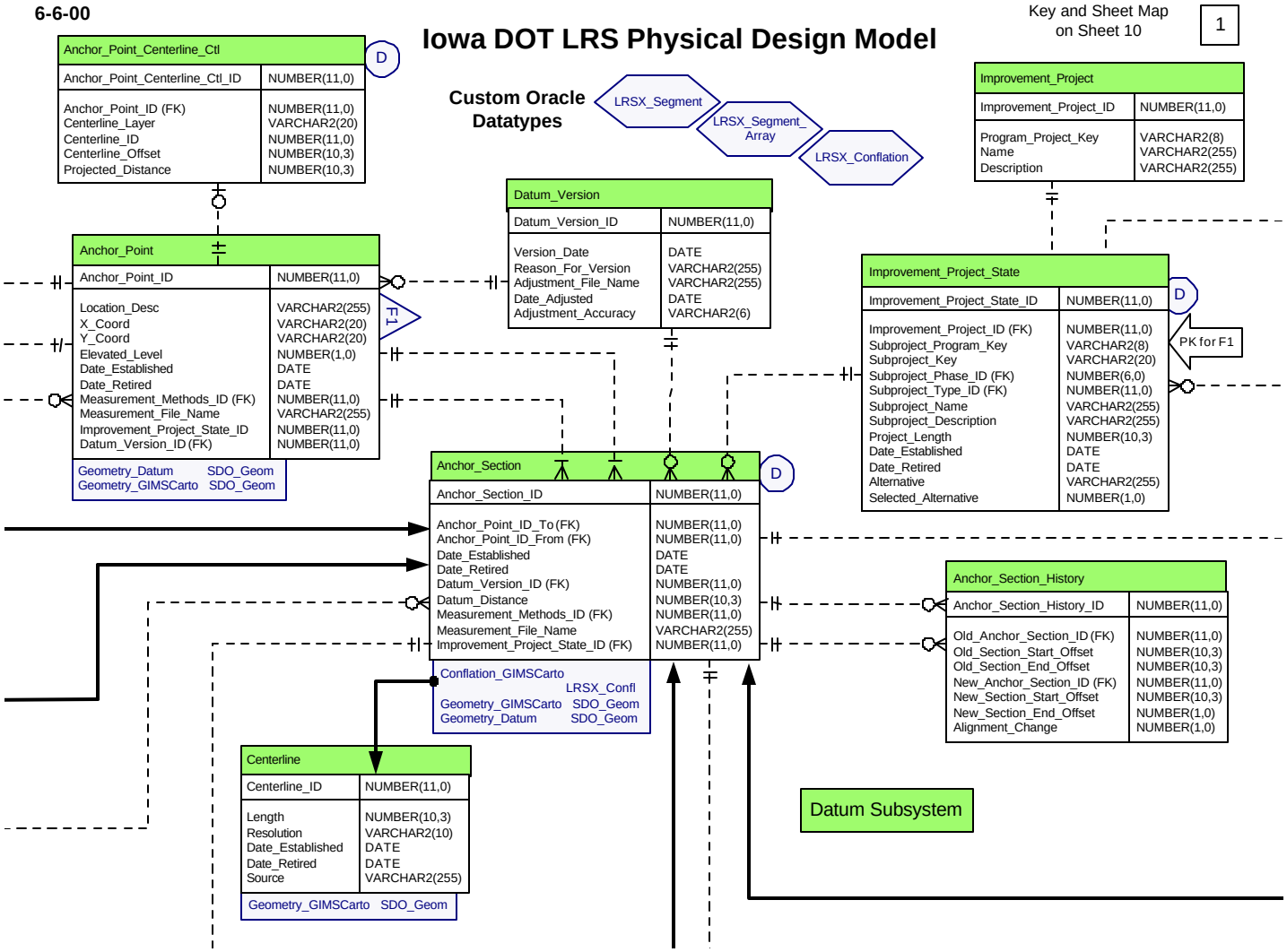


Figure 6: Diagram – Route Subsystem – Network Tables (Sheet 2)

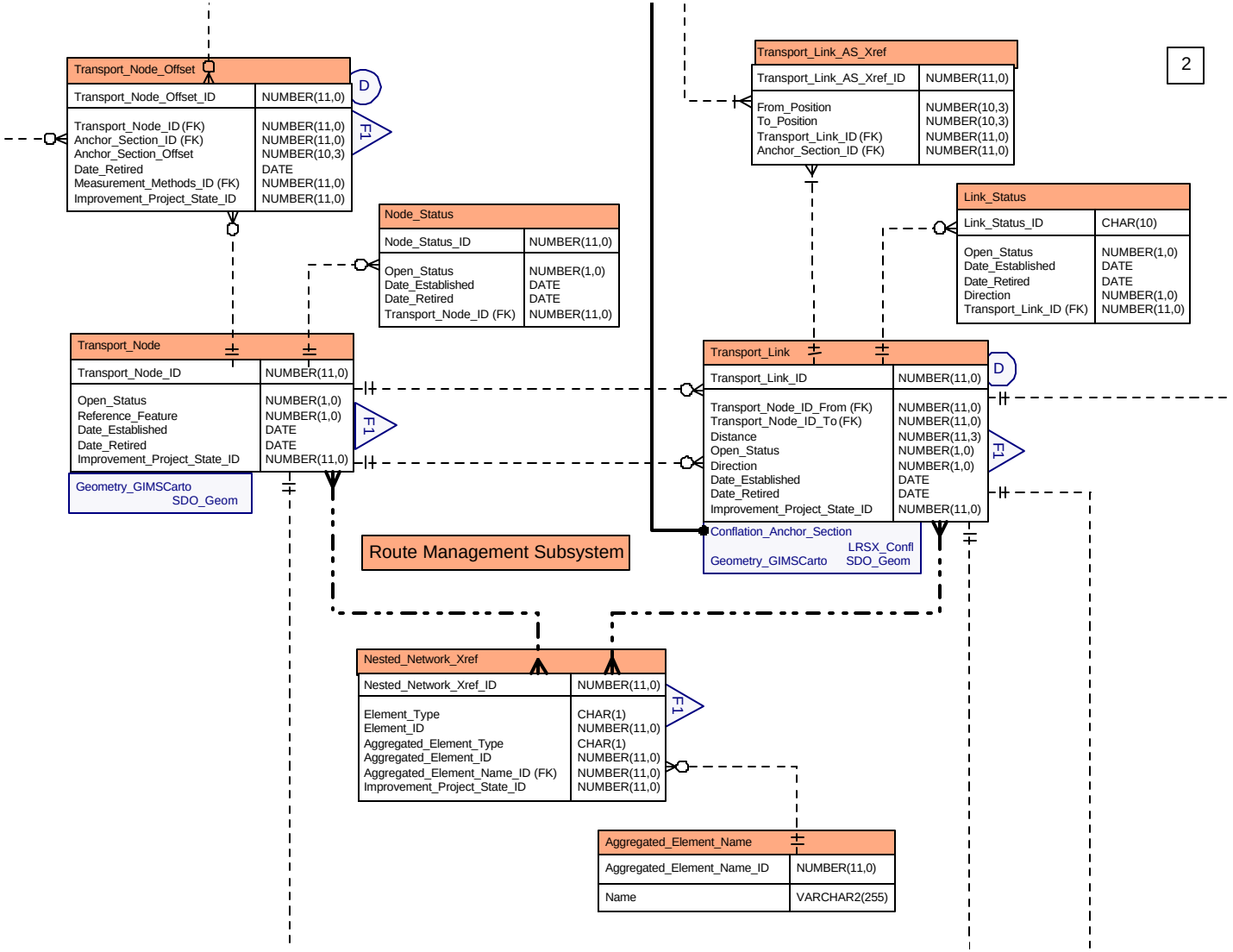


Figure 7: Diagram – Literal Description LRM Subsystem and Route Subsystem Route Tables (Sheet 3)

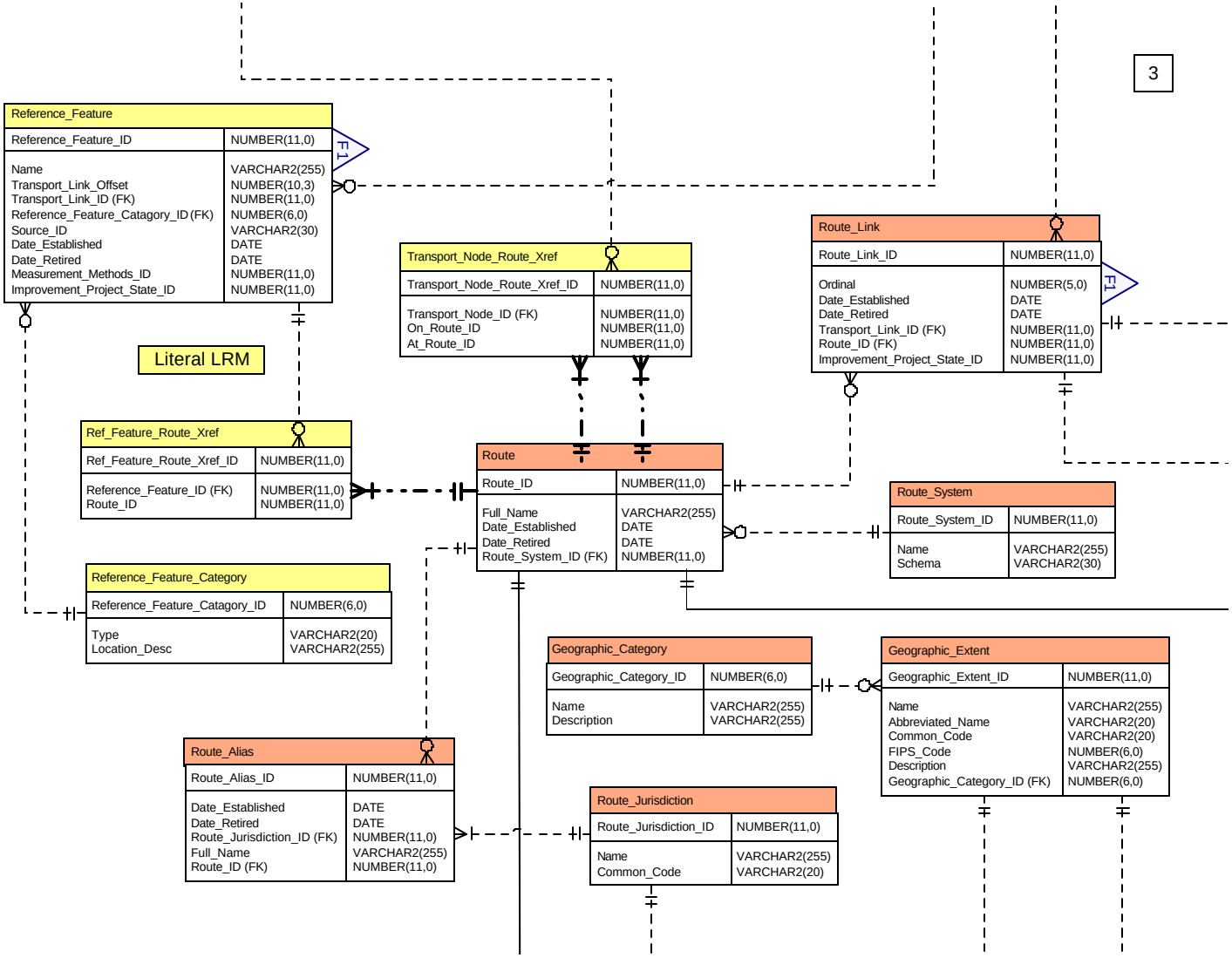


Figure 8: Diagram – Individual Route Systems (Sheet 4)

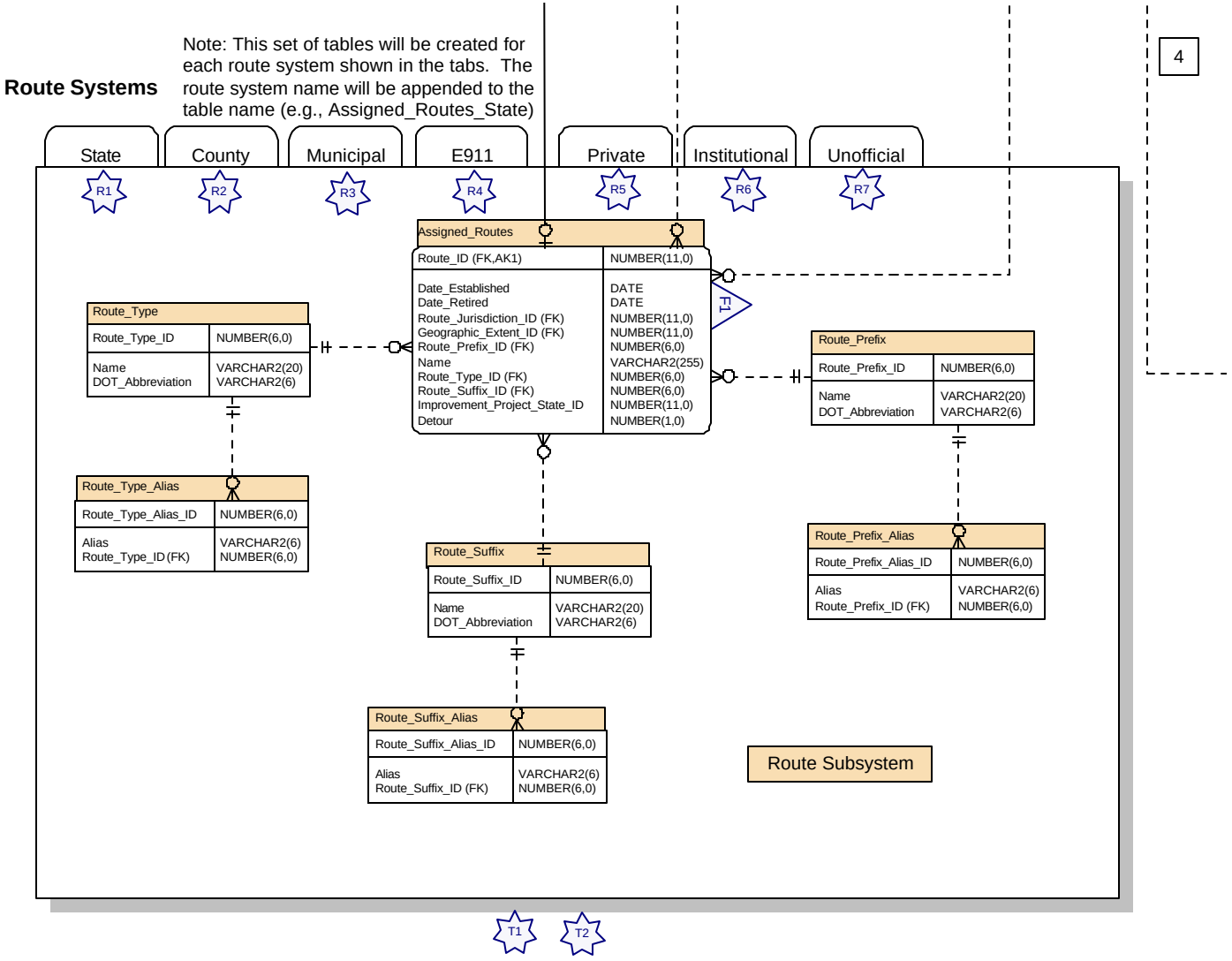


Figure 9: Diagram – Stationing LRM Subsystem and Datum Subsystem Tables (Sheet 5)

5

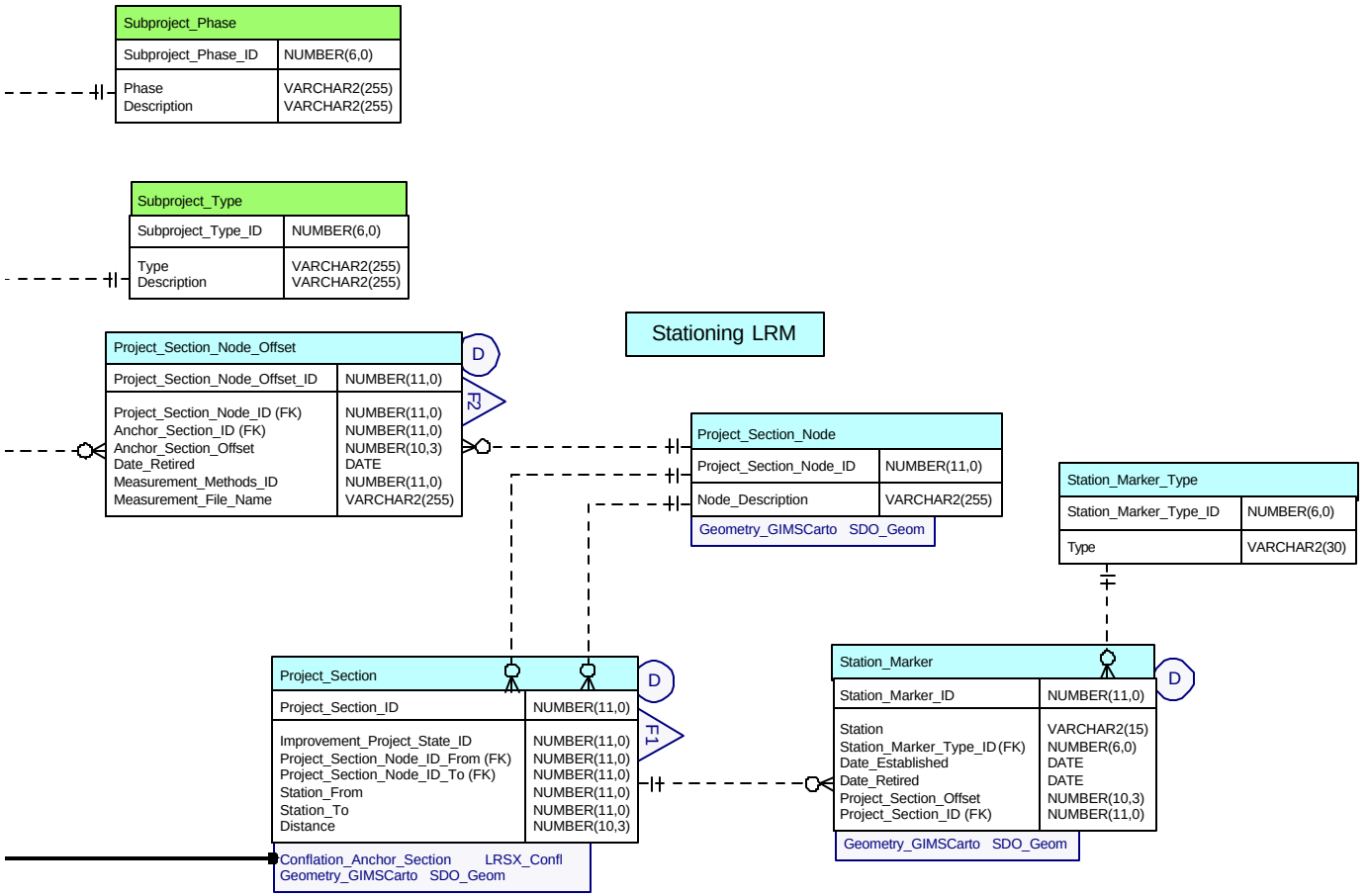


Figure 10: Diagram – Route Subsystem Transport System Tables (Sheet 6)

6

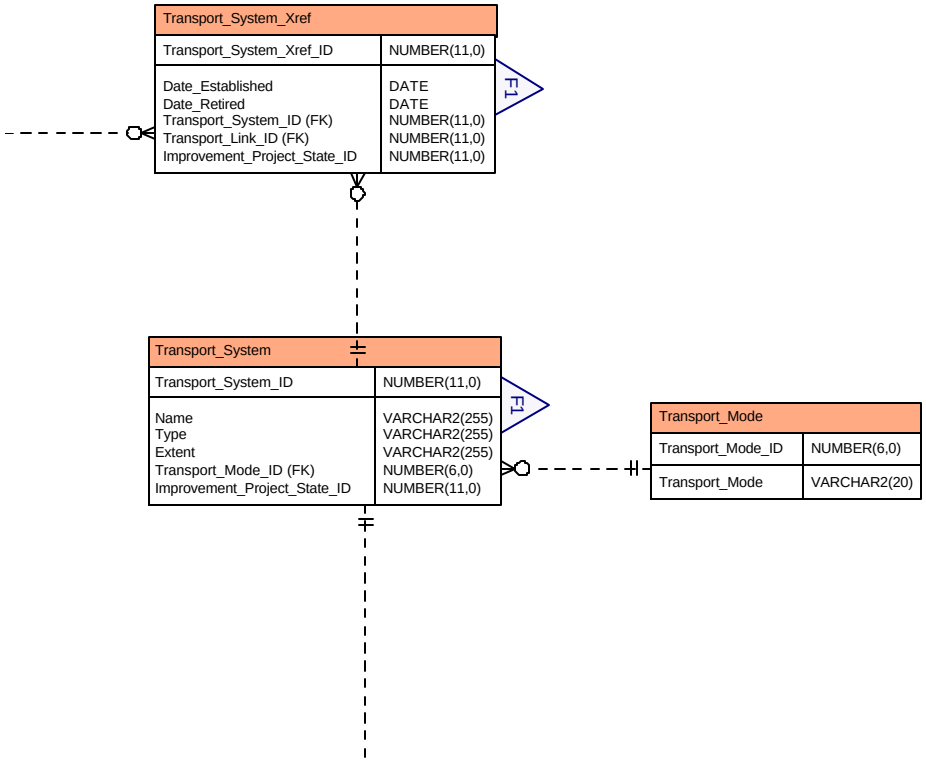


Figure 11: Diagram – Reference Post LRM Subsystem and Milepoint LRM Table (Sheet 7)

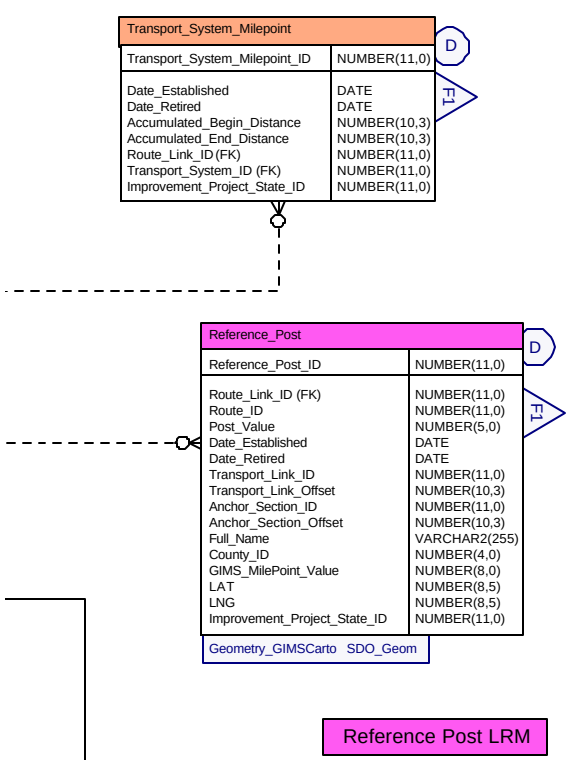


Figure 12: Diagram – Ramp Route System Tables (Sheet 8)

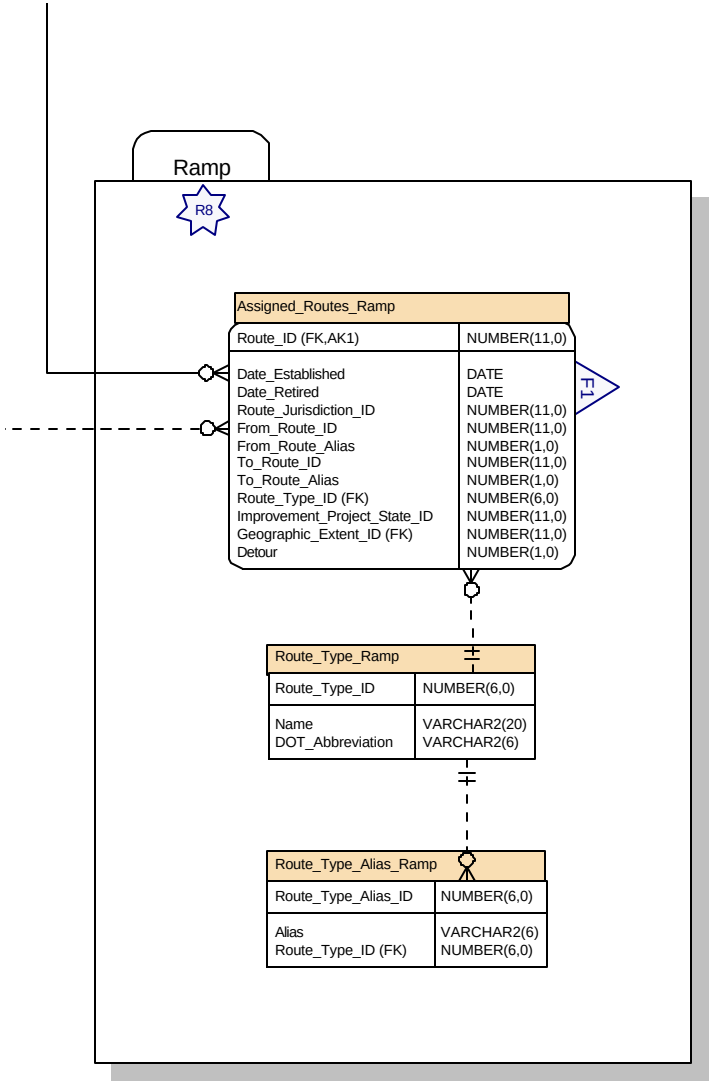


Figure 13: Diagram – Segmental LRM Subsystem and Datum Subsystem Tables (Sheet 9)

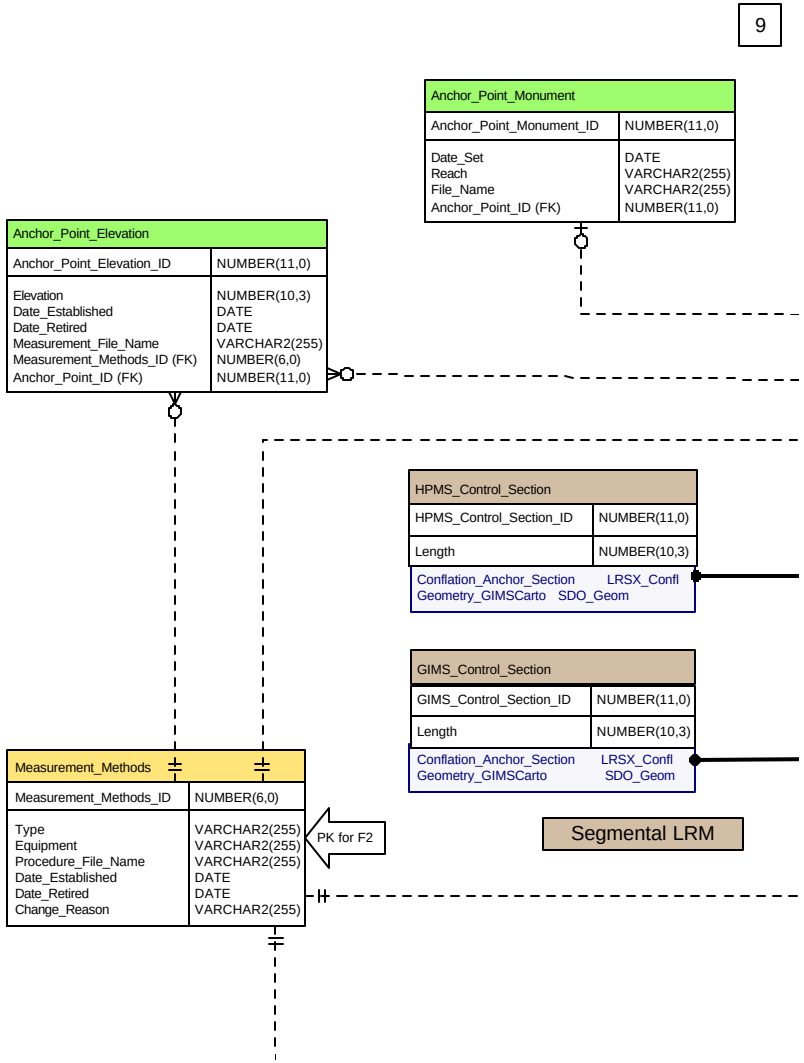
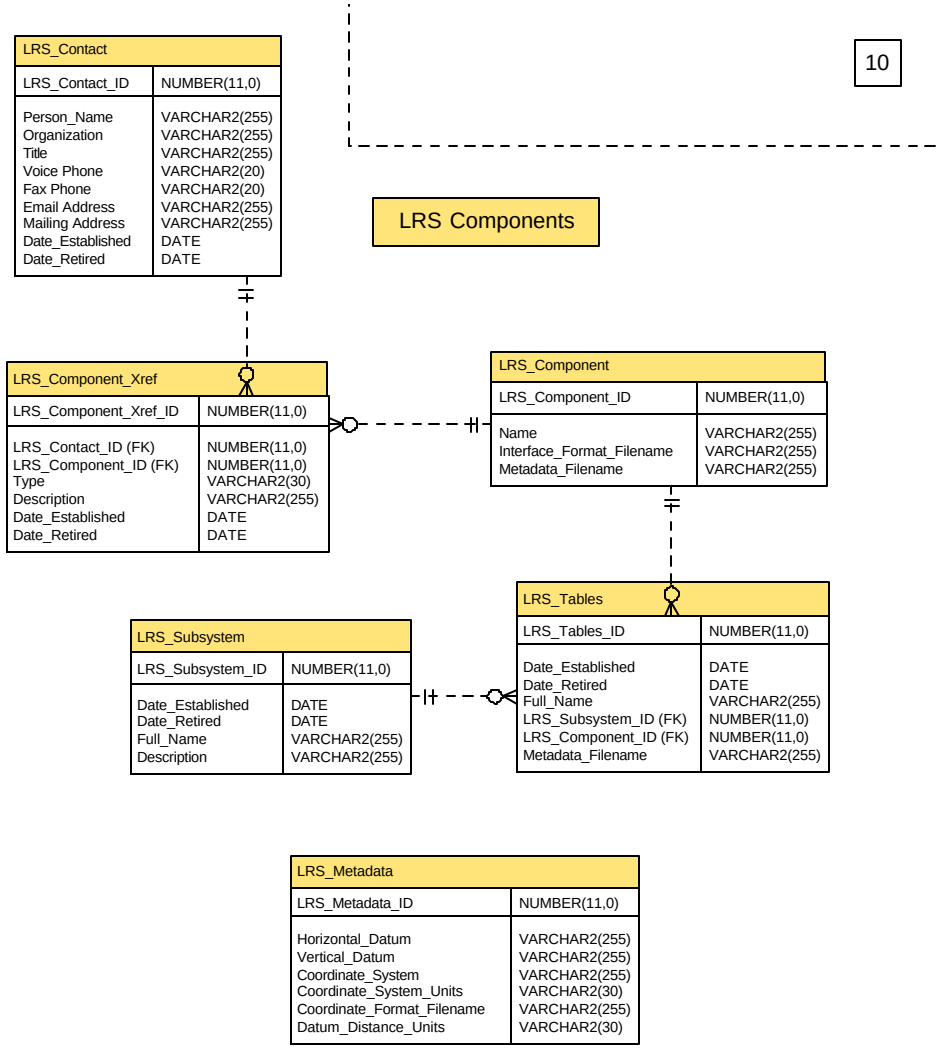


Figure 14: Diagram – LRS Component Tables (Sheet 10)



3 DATA DICTIONARY

3.1 Background

As mentioned above, Visio has the capability of producing a report of the database entities that are defined in a document model. The report is a text version of the information contained in the diagram. There are many options that can be selected to determine what the report contains.

The Physical Design Team used the "Notes" field associated with diagram entities to store definitions of the elements. Project participants can then incorporate this field in the report. This allows the Team to maintain a very complete and concise representation of the model within one document. Project participants can then have the option of presenting the model in a variety of ways, graphically and textually, in order to more fully understand the model.

The following presentation of the model is just one way the definitions can be reported. The report is organized by table entity, in alphabetical order. After an overall description of the table, the primary key is identified and then followed by a list of all the columns and their datatypes. Foreign keys are indicated by the "(FK)" notation and then described more completely at the end of the table listing. The column list is followed by detailed definitions of what the values in the columns represent within the model. At the end are the foreign key descriptions, containing parent tables, cardinality and referential integrity settings.

The main body of the report begins on the following page. The report's main body is divided into two sections based on Visio's inability to represent, and therefore report on, Oracle extended object data types: 3.2 Tables and Columns of Standard Datatypes and 3.3 Tables and Columns of Extended Data Types.

3.2 Tables and Columns of Standard Datatypes

Iowa DOT LRS Operational Database

Physical Design - Data Dictionary 6-6-00

Aggregated_Element_Name

Notes: A node or link that is a complex object composed of two or more other nodes and links. That is, an element can either a node or a link and either can be composed of both nodes and links.

Primary key: Aggregated_Element_Name_ID

Columns	Data type	Allow NULLs	Value/Range
Aggregated_Element_Name_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	

Column details

1. Aggregated_Element_Name_ID

Notes: Primary key which is a unique and arbitrary value.

2. Name

Notes: The name of the element. The name must be spelled out fully and make sense to the average public. Codes or acronyms can be included but only in parathenses at the end of the name. Examples of valid names are "The I-80 and I35 Interchange North of Des Moines, Iowa', or the 'The North-South Roadway Corridor between Ames and Des Moines, Iowa".

Anchor_Point

Notes: "...a zero-dimensional location that can be uniquely identified in the real world in such a way that its position can be determined and recovered in the field. Each anchor point has a 'location description' attribute..."

Primary key: Anchor_Point_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Point_ID	NUMBER(11,0)	Not allowed	
Location_Desc	VARCHAR2(255)	Not allowed	
X_Coord	VARCHAR2(20)	Not allowed	

Y_Coord	VARCHAR2(20)	Not allowed
Elevated_Level	NUMBER(1,0)	Allowed
Date_Established	DATE	Not allowed
Date_Retired	DATE	Allowed
Measurement_Methods_ID (FK)	NUMBER(11,0)	Not allowed
Measurement_File_Name	VARCHAR2(255)	Not allowed
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed
Datum_Version_ID (FK)	NUMBER(11,0)	Not allowed

Column details

1. Anchor_Point_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Location_Desc
Notes: Provides the information necessary for determining and recovering the anchor point's position in the field. The information is either quantitative or descriptive locators or both.

3. X_Coord
Notes: This is the X Coordinate value or easting for a coordinate set . This may be measured in latitude coordinates, for example. The value represents the most accurate and precise value the DOT has of its true location.

4. Y_Coord
Notes: This is the Y coordinate value or northing for a coordinate set. This may be measured in longitude coordinates, for example. The value represents the most accurate and precise value the DOT has of its true location.

5. Elevated_Level
Notes: Designation of 'Upper', 'Lower', or 'Middle'. Represents ordinal height of anchor points that share common (x,y) space but not elevation. This is to accomoate overpasses - it is an optional attribute.

6. Date_Established
Notes: The date the Anchor Point is established in the field.

7. Date_Retired
Notes: The date that reflects the Improvement Project State retired date. If the anchor point has a state of zeo, it is the date that the anchor point no longer exists in the real world, typcially driven by an improvement project that obliterates the location.

8. Measurement_Methods_ID (FK)
Notes: Foreign key relating to the Measurement Methods entity type.

9. Measurement_File_Name
Notes: A text string to point to an external file or database that contains the originally collected information.

10. Improvement_Project_State_ID
Notes: Foreign key relating to the Improvement Project State entity type.

11. Datum_Version_ID (FK)
Notes: Foreign key relating to the Datum Version entity type.

Foreign key details (child)

Msrmt_Methods_Anch_Pt_FK

Definition:	**Child**	**Parent**
	Measurement_Methods_ID	Measurement_Methods. Measurement_Methods_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Datum_Version_Anchor_Point_FK

Definition:	**Child**	**Parent**
	Datum_Version_ID	Datum_Version.Datum_Version_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anchor_Point_Centerline_Ctl

Notes: A projection of the anchor point onto a cartographic representation of space. Each anchor point must be snapped to a cartographic element and this table records the results of the snapping process.

Primary key: Anchor_Point_Centerline_Ctl_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Point_Centerline_Ctl_ID	NUMBER(11,0)	Not allowed	
Anchor_Point_ID (FK)	NUMBER(11,0)	Not allowed	
Centerline_Layer	VARCHAR2(20)	Not allowed	
Centerline_ID	NUMBER(11,0)	Not allowed	
Centerline_Offset	NUMBER(10,3)	Not allowed	
Projected_Distance	NUMBER(10,3)	Not allowed	

Column details

1. Anchor_Point_Centerline_Ctl_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Anchor_Point_ID (FK)
Notes: Foreign key relating to the Anchor Point entity type.

3. Centerline_Layer
Notes: The system name (schema and table name) of the cartographic system used as the snap source.

4. Centerline_ID
Notes: Foreign key into the Centerline table referencing the centerline feature that the anchor point is snapped to.

5. Centerline_Offset
Notes: Real valued offset calculated in system units (meters) from the start of the cartographic features line string to the snap point.

6. Projected_Distance
Notes: Orthogonal distance in system units (meters) between the original surveyed location for the anchor point and the snap point on the cartographic element.

Foreign key details (child)

Anch_Pt_Anch_Pt_Ctrln_Ctrl_FK

	Child	Parent
Definition:	Anchor_Point_ID	Anchor_Point.Anchor_Point_ID
Cardinality:	One -to- Zero-or-One	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Anchor_Point_Elevation

Notes: The absolute elevation (Z values) of the anchor point.

The elevation value is separate because its purpose is more long term and is less defined than an anchor point's X,Y coordinate values. Its value will change frequently in the next few years as improved accuracies are acquired.

Primary key: Anchor_Point_Elevation_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Point_Elevation_ID	NUMBER(11,0)	Not allowed	
Elevation	NUMBER(10,3)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Measurement_File_Name	VARCHAR2(255)	Not allowed	
Measurement_Methods_ID (FK)	NUMBER(6,0)	Not allowed	
Anchor_Point_ID (FK)	NUMBER(11,0)	Not allowed	

Column details

1. Anchor_Point_Elevation_ID

Notes: Primary key that is a unique arbitrary value for each record in the entity.

2. Elevation

Notes: The actual Z value in geodetic measurements.

3. Date_Established

Notes: The date when the elevation was measured (if it was processed, the date it was processed).

4. Date_Retired

Notes: The date when the elevation is retired because a more accurate elevation is obtained.

5. Measurement_File_Name

Notes: A text string that contains the path to an external file that contains the originally collected information.

6. Measurement_Methods_ID (FK)

Notes: Foreign key to relate to the Measurement Methods entity type.

7. Anchor_Point_ID (FK)
Notes: Foreign Key that links monument to the anchor point.

Foreign key details (child)
Anch_Pt_Anch_Pt_Elevation_FK

	Child	Parent
Definition:	Anchor_Point_ID	Anchor_Point.Anchor_Point_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Msrmt_Methods_Anch_Pt_Elev_FK

	Child	Parent
Definition:	Measurement_Methods_ID	Measurement_Methods. Measurement_Methods_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anchor_Point_Monument

Notes: A physical device that witnesses the location of an Anchor Point. Monuments are used as to identify Anchor Points which cannot easily be identified and recovered in the field. Witness means that the anchor point may not be at the same location as the Monunment. For example, the monument could be a physical post in the right-of-way.

Primary key: Anchor_Point_Monument_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Point_Monument_ID	NUMBER(11,0)	Not allowed	
Date_Set	DATE	Not allowed	
Reach	VARCHAR2(255)	Not allowed	
File_Name	VARCHAR2(255)	Not allowed	
Anchor_Point_ID (FK)	NUMBER(11,0)	Not allowed	

Column details
1. Anchor_Point_Monument_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Date_Set
Notes: Date monument was set.

3. Reach
Notes: Text field that describes how to find the anchor point monument in the field, what it looks like.

4. File_Name

Notes: Link to external file containing specifications of monument collection of data.

5. Anchor_Point_ID (FK)
Notes: Foreign Key that links monument to the anchor point.

Foreign key details (child)
Anch_Pt_Anch_Pt_Monument_FK

Definition:	**Child**	**Parent**
	Anchor_Point_ID	Anchor_Point.Anchor_Point_ID

Cardinality: One -to- Zero-or-One
Allow NULLs: Not allowed
Ref. Integrity on update: Cascade
Ref. Integrity on delete: Cascade

Anchor_Section

Notes: A stable and elementary part of the transportation linear reference space. "...a continuous, directed, non-branching linear feature, connecting two anchor points, whose real-world length is measured in the field."

Primary key: Anchor_Section_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Section_ID	NUMBER(11,0)	Not allowed	
Anchor_Point_ID_To (FK)	NUMBER(11,0)	Not allowed	
Anchor_Point_ID_From (FK)	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Datum_Version_ID (FK)	NUMBER(11,0)	Not allowed	
Datum_Distance	NUMBER(10,3)	Not allowed	
Measurement_Methods_ID (FK)	NUMBER(11,0)	Not allowed	
Measurement_File_Name	VARCHAR2(255)	Not allowed	
Improvement_Project_State_ID (FK)	NUMBER(11,0)	Not allowed	

Column details
1. Anchor_Section_ID
Notes: Primary key that is a unique and arbitrary number for each record in the entity.

2. Anchor_Point_ID_To (FK)
Notes: Identifies the Anchor Point that defines the ending of an Anchor Section.

3. Anchor_Point_ID_From (FK)
Notes: Identifies the Anchor Point that defines the beginning of an Anchor Section.

4. Date_Established
Notes: The date that the Anchor Section is available to others to use. At first, this is the same date as the Datum Version. If a section is added after a version is created, it is most likely the date that the anchor section's distance is determined.

5. Date_Retired

Notes: The date that reflects the Improvement Project State retired date. If the anchor section has a state of zero, it is the date that the anchor section no longer exists in the real world, typcially driven by an improvement project that obliterates any or all of the anchor section's location.

6. Datum_Version_ID (FK)
Notes: Foreign key relating to the Datum Version entity type.

7. Datum_Distance
Notes: CALCULATED: the distance for the section determined from collected measurements. Depending on the measurement source and method of collection, some measurements will be run through an error adjustment.

8. Measurement_Methods_ID (FK)
Notes: A foreign key that associates the method used to collect the datum distances.

9. Measurement_File_Name
Notes: A text string that points to an external file or database that contains the originally collected information. This file includes all the redundancy length information that is used to perform the adjustment and create the datum distance.

10. Improvement_Project_State_ID (FK)
Notes: A foreign key that associates the Anchor Section with a construction project date that obliterates the roadway the Anchor Section represents. This date will allow data users the ability to select only "current" Anchor Sections.

Foreign key details (child)

Datum_Version_Anch_Section_FK

Definition:	**Child**	**Parent**
	Datum_Version_ID	Datum_Version.Datum_Version_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anchor_Point_ID_To_FK

Definition:	**Child**	**Parent**
	Anchor_Point_ID_To	Anchor_Point.Anchor_Point_ID

Cardinality: One -to- One-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anchor_Point_ID_From_FK

Definition:	**Child**	**Parent**
	Anchor_Point_ID_From	Anchor_Point.Anchor_Point_ID

Cardinality: One -to- One-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Imp_Proj_State_Anch_Sec_FK

Definition:	**Child**	**Parent**
	Improvement_Project_State_ID	Improvement_Project_State. Improvement_Project_State_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Msrmt_Methods_Anch_Section_FK

Definition:	**Child**	**Parent**
	Measurement_Methods_ID	Measurement_Methods. Measurement_Methods_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anchor_Section_History

Notes: The spatial (linear) relationship between anchor sections that share the same location but at different points in time. This information is necessary to reattach business data from portions of a retired section that are still valid, (ie: the portion where the roadway alignment has not changed) to the new section replacing it and to allow business data this is impacted by minor datum adjustments the ability to attach to the most current datum location.

Primary key: Anchor_Section_History_ID

Columns	Data type	Allow NULLs	Value/Range
Anchor_Section_History_ID	NUMBER(11,0)	Not allowed	
Old_Anchor_Section_ID (FK)	NUMBER(11,0)	Not allowed	
Old_Section_Start_Offset	NUMBER(10,3)	Not allowed	
Old_Section_End_Offset	NUMBER(10,3)	Not allowed	
New_Anchor_Section_ID (FK)	NUMBER(11,0)	Not allowed	
New_Section_Start_Offset	NUMBER(10,3)	Not allowed	
New_Section_End_Offset	NUMBER(1,0)	Not allowed	
Alignment_Change	NUMBER(1,0)	Not allowed	

Column details

1. Anchor_Section_History_ID
Notes: Primary key that is a unique and arbitrary number for each record in the entity.

2. Old_Anchor_Section_ID (FK)
Notes: The identifier of the anchor section that is being retired.

3. Old_Section_Start_Offset
Notes: The distance (in meters) along the old section where the new section begins, in the direction of the old section From Anchor Point and To Anchor Point.

4. Old_Section_End_Offset

Notes: The distance (in meters) along the old section where the new section end, in the direction of the old section From Anchor Point and To Anchor Point.

5. New_Anchor_Section_ID (FK)
Notes: The identifier of the anchor section that is being added.

6. New_Section_Start_Offset
Notes: The distance (in meters) along the new section where the old section begins, in the direction of the new section From Anchor Point and To Anchor Point.

7. New_Section_End_Offset
Notes: The distance (in meters) along the new section where the old section ends, in the direction of the new section From Anchor Point and To Anchor Point.

8. Alignment_Change
Notes: Whether this particular relationship is due to a real world change (1) or due only because the anchor section ID needed to be retired because another portion of the section was modified by a real world change (0).

Foreign key details (child)
Old_Anchor_Section_ID_FK

Definition:	**Child**	**Parent**
	Old_Anchor_Section_ID	Anchor_Section.Anchor_Section_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

New_Anchor_Section_ID_FK

Definition:	**Child**	**Parent**
	New_Anchor_Section_ID	Anchor_Section.Anchor_Section_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Assigned_Routes

Notes: An ordered and directed, but not necessarily connected, set of whole Transport Links (based on Traversal object from NCHRP 20-27(2)). These can be posted routes in the field, inventory routes, snow-plow routes, transit routes, scenic road routes, etc.

Primary key:

Columns	Data type	Allow NULLs	Value/Range
Route_ID (FK,U1)	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	

Route_Jurisdiction_ID (FK)	NUMBER(11,0)	Not allowed
Geographic_Extent_ID (FK)	NUMBER(11,0)	Not allowed
Route_Prefix_ID (FK)	NUMBER(6,0)	Allowed
Name	VARCHAR2(255)	Not allowed
Route_Type_ID (FK)	NUMBER(6,0)	Allowed
Route_Suffix_ID (FK)	NUMBER(6,0)	Allowed
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed
Detour	NUMBER(1,0)	Not allowed

Indexes	Columns	Sort order
Assigned_Route_Alt_I (U1)	Route_ID	Ascending

Column details

1. Route_ID (FK,U1)

2. Date_Established
Notes: The date/time when the route was officially made a route by the governing body of the route system (legislature, city council, DOT, etc).

3. Date_Retired
Notes: The date/time when the route was officially retired by the governing body of the route system (legislature, city council, DOT, etc).

4. Route_Jurisdiction_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Jurisdiction table.

5. Geographic_Extent_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Geographic Extent table.

6. Route_Prefix_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Prefix table.

7. Name
Notes: A text field to hold the "official" name of the route. For example "30" for US 30 East, or "Main" for North Main Street.

8. Route_Type_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Type table.

9. Route_Suffix_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Suffix table.

10. Improvement_Project_State_ID
Notes: A foreign key that must be exactly the same attribute as in the Improvement Project State table.

11. Detour
Notes: A boolean indicating whether the route is a detour (1) or not (0).

Index details

Assigned_Route_Alt_I
Column(s): Route_ID (Asc)
Unique: Yes

Foreign key details (child)

Rt_Type_Assigned_Routes_FK

Definition:	**Child** Route_Type_ID	**Parent** Route_Type.Route_Type_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Rr_Suffix_Assigned_Routes_FK

Definition:	**Child** Route_Suffix_ID	**Parent** Route_Suffix.Route_Suffix_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Route_Assigned_Routes_FK

Definition:	**Child** Route_ID	**Parent** Route.Route_ID
Cardinality:	One -to- Zero-or-One	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Rt_Prefix_Assigned_Routes_FK

Definition:	**Child** Route_Prefix_ID	**Parent** Route_Prefix.Route_Prefix_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Geog_Extent_Assigned_Routes_FK

Definition:	**Child** Geographic_Extent_ID	**Parent** Geographic_Extent.Geographic_Extent_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Route_Juris_Assigned_Routes_FK

Definition:	**Child** Route_Jurisdiction_ID	**Parent** Route_Jurisdiction.Route_Jurisdiction_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	

Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Assigned_Routes_Ramp

Notes: An ordered and directed, but not necessarily connected, set of whole Transport Links (based on Traversal object from NCHRP 20-27(2)) that describe a transition from one route to another.

Primary key:

Columns	Data type	Allow NULLs	Value/Range
Route_ID (FK,U1)	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Route_Jurisdiction_ID	NUMBER(11,0)	Not allowed	
From_Route_ID	NUMBER(11,0)	Allowed	
From_Route_Alias	NUMBER(1,0)	Allowed	
To_Route_ID	NUMBER(11,0)	Allowed	
To_Route_Alias	NUMBER(1,0)	Allowed	
Route_Type_ID (FK)	NUMBER(6,0)	Allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	
Geographic_Extent_ID (FK)	NUMBER(11,0)	Not allowed	
Detour	NUMBER(1,0)	Not allowed	

Indexes	Columns	Sort order
Assigned_Rt_Ramp_Alt_I (U1)	Route_ID	Ascending

Column details

1. Route_ID (FK,U1)
Notes: Route_ID identifies Assigned_Routes

2. Date_Established
Notes: The date/time when the route was officially made a route by the governing body of the route system (legislature, city council, DOT, etc).

3. Date_Retired
Notes: The date/time when the route was officially retired by the governing body of the route system (legislature, city council, DOT, etc).

4. Route_Jurisdiction_ID
Notes: A foreign key that must be exactly the same attribute as in the Route Jurisdiction table. This the body that officially is responsible for naming the roadway (it provides the establish and retire dates).

5. From_Route_ID
Notes: The ID of the route from which traffic flows along the roadway. It is an ID from the Route or Route_Alias tables, NOT it's own Prefix and Alias tables. Because it can go to either table, this is not a Foreign Key.

6. From_Route_Alias
Notes: A boolean that indicates whether the From_Route_ID is from the Alias table (True) or from the Route table (False). The boolean is represented as 0 for True and 1 for False.

7. To_Route_ID
Notes: The ID of the route to which traffic flows along the roadway. It would use an ID from the Route or Route_Alias tables, NOT it's own Suffix and Alias tables. Because it can go to either table, this is not a Foreign Key.

8. To_Route_Alias
Notes: A boolean that indicates whether the To_Route_ID is from the Alias table (True) or from the Route table (False). The boolean is represented as 0 for True and 1 for False.

9. Route_Type_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Type table.

10. Improvement_Project_State_ID
Notes: A foreign key that must be exactly the same attribute as in the Improvement Project State table.

11. Geographic_Extent_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Geographic Extent table.

12. Detour
Notes: A boolean indicating whether the route is a detour (1) or not (0).

Index details

Assigned_Rt_Ramp_Alt_I
Column(s): Route_ID (Asc)
Unique: Yes

Foreign key details (child)

Rt_Type$Ramp_Assign_Rt$Ramp_FK

	Child	Parent
Definition:	Route_Type_ID	Route_Type_Ramp.Route_Type_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Geog_Ext_Assign_Routes$Ramp_FK

	Child	Parent
Definition:	Geographic_Extent_ID	Geographic_Extent.Geographic_Extent_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Rt_Assigned_Routes$Ramp_FK

	Child	Parent
Definition:	Route_ID	Route.Route_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed

Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Centerline

Notes: A centerline is equivalent to the Spatial Data Transfer Standard object called a string: "A connected non-branching sequence of line segments specified as the ordered sequence of points between those line segments. Note: A string may intersect itself or other strings (A FGDC Spatial Data Transfer Standard 'String', Section 2.3.2.2). " The centerline can represent any transport feature (roadway, rail, waterway, pedestrian way, etc.). The centerline can be created based a any compilation method: cartographic principles (generalization), remotely-sensed principles (satellite imagery, Aerial photography, planimetrics, orthometrics), parametric derivation (COGO), etc.

Primary key: Centerline_ID

Columns	Data type	Allow NULLs	Value/Range
Centerline_ID	NUMBER(11,0)	Not allowed	
Length	NUMBER(10,3)	Not allowed	
Resolution	VARCHAR2(10)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Not allowed	
Source	VARCHAR2(255)	Not allowed	

Column details

1. Centerline_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Length
Notes: Distance of the string in spatial units of measure.

3. Resolution
Notes: The resolution (imagery) or scale (cartographic data) from which the string was derived.

4. Date_Established
Notes: The date the real world object(s) that the cartography represents is established.

5. Date_Retired
Notes: The date the real world object(s) that the cartography represents is retired.

6. Source
Notes: A text string that describes the method used to collect the string, or the existing data set name from which the string was extracted.

Datum_Version

Notes: The set of anchor sections to which a mathematical distance adjustment was made. These adjustments are global adjustments, meaning most if not all distances may be modified.

Primary key: Datum_Version_ID

Columns	Data type	Allow NULLs	Value/Range
Datum_Version_ID	NUMBER(11,0)	Not allowed	
Version_Date	DATE	Not allowed	
Reason_For_Version	VARCHAR2(255)	Not allowed	
Adjustment_File_Name	VARCHAR2(255)	Not allowed	
Date_Adjusted	DATE	Not allowed	
Adjustment_Accuracy	VARCHAR2(6)	Not allowed	

Column details

1. Datum_Version_ID

Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Version_Date

Notes: The date of the datum version to which the LRS is attached.

3. Reason_For_Version

Notes: A text field that provides a synopsis of why it was necessary to create a new version and the benefits the new version will provide.

4. Adjustment_File_Name

Notes: Field contains the name and location of the external file(s) that contain the data that stipulated an adjustment was needed such as the Mandli file name, least square adjustment file name, etc.

5. Date_Adjusted

Notes: The date of the adjustment was made.

6. Adjustment_Accuracy

Notes: The accuracy or "confidence level" of the adjustment.

Geographic_Category

Notes: An enumerated list of geographic extent types.

Primary key: Geographic_Category_ID

Columns	Data type	Allow NULLs	Value/Range
Geographic_Category_ID	NUMBER(6,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Description	VARCHAR2(255)	Not allowed	

Column details

1. Geographic_Category_ID

Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Name

Notes: The full name of the category. The current list: State, County, Municipality, Urban Area, Institution, DOT Transportation Center, Federal Highway

District.

3. Description
Notes: A description of the criteria for inclusion and a list of extent types included in this category.

Geographic_Extent

Notes: Any geographic extent with a name recognized by the DOT. This is not the owner or maintainer of a roadway. It is the intuitive assignment that most anyone using the LRS would conclude that a roadway is found.

Primary key: Geographic_Extent_ID

Columns	Data type	Allow NULLs	Value/Range
Geographic_Extent_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Abbreviated_Name	VARCHAR2(20)	Not allowed	
Common_Code	VARCHAR2(20)	Allowed	
FIPS_Code	NUMBER(6,0)	Not allowed	
Description	VARCHAR2(255)	Allowed	
Geographic_Category_ID (FK)	NUMBER(6,0)	Not allowed	

Column details

1. Geographic_Extent_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Name
Notes: The full name of the geography. For example, "Iowa State University - Ames Campus".

3. Abbreviated_Name
Notes: An abbreviated version of Name to be used as part of the a route name. For example, "ISU-Ames".

4. Common_Code
Notes: The DOT code already assigned to this extent.

5. FIPS_Code
Notes: The federal code.

6. Description
Notes: OPTIONAL: A general comment to help clarify the geography extent.

7. Geographic_Category_ID (FK)
Notes: Foreign key relating to the Geographic Category table.

Foreign key details (child)
Geog_Category_Geog_Extent_FK

Definition:	**Child**	**Parent**
	Geographic_Category_ID	Geographic_Category.Geographic_Category_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

GIMS_Control_Section

Notes: An entity type that describes the Base Record's piece of pavement. The roadway is segmented according to a set of business-driven criteria.

Primary key: GIMS_Control_Section_ID

Columns	Data type	Allow NULLs	Value/Range
GIMS_Control_Section_ID	NUMBER(11,0)	Not allowed	
Length	NUMBER(10,3)	Not allowed	

Column details

1. GIMS_Control_Section_ID
Notes: Primary key of a unique and arbitrary number.

2. Length
Notes: DERIVED: From the Anchor Section Datum Distance.

HPMS_Control_Section

Notes: An entity type that describes the HPMS piece of pavement. The roadway is segmented according to a set of business-driven criteria.

Primary key: HPMS_Control_Section_ID

Columns	Data type	Allow NULLs	Value/Range
HPMS_Control_Section_ID	NUMBER(11,0)	Not allowed	
Length	NUMBER(10,3)	Not allowed	

Column details

1. HPMS_Control_Section_ID
Notes: Primary key of a unique and arbitrary number.

2. Length
Notes: DERIVED: From the Anchor Section Datum Distance.

Improvement_Project

Notes: The program level identification and characteristics of each improvement project. An improvement project might be roadway improvement project, an extension of the datum, and error correction to the database, and datum adjustment, etc.

Primary key: Improvement_Project_ID

Columns	Data type	Allow NULLs	Value/Range
Improvement_Project_ID	NUMBER(11,0)	Not allowed	
Program_Project_Key	VARCHAR2(8)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Description	VARCHAR2(255)	Not allowed	

Column details

1. Improvement_Project_ID

Notes: Primary key that is a unique arbitrary number for each record in the entity (system defined).

2. Program_Project_Key

Notes: A text field that holds the project key of the overall project. For Roadway Improvement Projects, this is the Iowa DOT's roadway improvement project ID assigned to the project as reference to external files or databases. This will be the new PIN# equivalent - this value will need to be assigned earlier than the program, back in conceptual discussions.

A datum extension project might have a contract #, or an error correction might received some categorical naming convention.

3. Name

Notes: A name that would be obvious to those involved in the project. For roadway improvement projects, this would be DOT and the public; example names could be "West Ames Bypass" or "Hwy 30-Skunk River Bridge Deck Replacement".

4. Description

Notes: A general description of the project, primarily targeted for correspondence with DOT management, the legislature, or the public.

Improvement_Project_State

Notes: The condition of an overall improvement project or it's subprojects. For example, the particular phase of a roadway improvement project within the facility development life cycle, or the first phase of extending the LRS beyond the first implementation phase.

Primary key: Improvement_Project_State_ID

Columns	Data type	Allow NULLs	Value/Range
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	
Improvement_Project_ID (FK)	NUMBER(11,0)	Not allowed	
Subproject_Program_Key	VARCHAR2(8)	Not allowed	

Subproject_Key	VARCHAR2(20)	Not allowed
Subproject_Phase_ID (FK)	NUMBER(6,0)	Not allowed
Subproject_Type_ID (FK)	NUMBER(11,0)	Not allowed
Subproject_Name	VARCHAR2(255)	Not allowed
Subproject_Description	VARCHAR2(255)	Not allowed
Project_Length	NUMBER(10,3)	Not allowed
Date_Established	DATE	Not allowed
Date_Retired	DATE	Allowed
Alternative	VARCHAR2(255)	Not allowed
Selected_Alternative	NUMBER(1,0)	Not allowed

Column details

1. Improvement_Project_State_ID
Notes: Primary key that is a unique arbitrary number for each record in the entity (system defined).

2. Improvement_Project_ID (FK)
Notes: Foreign key relating to the Improvement Project entity type.

3. Subproject_Program_Key
Notes: A text field that holds the Iowa DOT's roadway improvement project ID assigned to the project or subproject as reference to external files or databases. This is where the smart key PIN# would be used with sequences and levels.

4. Subproject_Key
Notes: A code used to refer to this project relative to the overall project. For roadway improvement projects, it would be the Sequencing ID assigned a portion of the overall project (e.g., a section of roadway), simple numbers, unique to a given overall project. (e.g., 00-20). This value is zero when the overall project does not break down into subprojects.

5. Subproject_Phase_ID (FK)
Notes: Foreign key relating to the Subproject Phase entity type.

6. Subproject_Type_ID (FK)
Notes: Foreign key relating to the Subproject Type entity type.

7. Subproject_Name
Notes: A name that would be indentifiable to all those who need to be informed of the project. For roadway improvement projects, the name needs to be familar to the DOT and the public. This value would remain the same as the Improvement Project Name attribute in the Improvement Project table if the project has no subprojects.

8. Subproject_Description
Notes: A general description of the subproject, primarily targeted for correspondence with DOT management, the legislature, and the public. This value would remain the same as the Improvement Project Name attribute in the Improvement Project table if the project has no subprojects.

9. Project_Length
Notes: This attribute is requried only for roadway improvement projects. It is the overall linear distance for the project (i.e., 2.3 miles of roadway to be resurfaced). Required for LRM stationing project types.

10. Date_Established
Notes: A calendar date for the beginning of the project state. For a roadway

construction project, this would be the legal date in which construction contractors are authorized to proceed.

11. Date_Retired
Notes: A calendar date for the end of the project state.

12. Alternative
Notes: A name assigned to each of the different alternatives for a specific phase of an improvement project. If there are no alternatives, the value must be the same as the attribute Subproject_Name. If alternatives are itemized (e.g., 1,2,3 or A,B,C) the syntax for this attribute should be: "Alternative" + <itemized value> + ": " + <name>. For example: "Alternative B: West Edge of Eddie's Marsh". If alternatives are not itemized, the syntax for this attribute is <name>. For example: "West Edge of Eddie's Marsh".

13. Selected_Alternative
Notes: The alternative(s) that were selected during a specific phase of an improvement project that will move forward to the next phase. All records are set to FALSE by default until set to TRUE by the user. TRUE means the alternative was selected. For roadway improvement projects, the number of alternatives that are selected is expected to decrease as the project moves from early phases to later phases.

Foreign key details (child)

Subproj_Phase_Imp_Proj_St_FK

Definition:	**Child** Subproject_Phase_ID	**Parent** Subproject_Phase.Subproject_Phase_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Subproj_Type_Imp_Proj_St_FK

Definition:	**Child** Subproject_Type_ID	**Parent** Subproject_Type.Subproject_Type_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Imp_Proj_Imp_Proj_State_FK

Definition:	**Child** Improvement_Project_ID	**Parent** Improvement_Project.Improvement_Project_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Link_Status

Notes: At a given point in time, the change in condition of whether vehicles can travel along a link.

Primary key: Link_Status_ID

Columns	Data type	Allow NULLs	Value/Range
Link_Status_ID	CHAR(10)	Not allowed	
Open_Status	NUMBER(1,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Direction	NUMBER(1,0)	Not allowed	
Transport_Link_ID (FK)	NUMBER(11,0)	Not allowed	

Column details

1. Link_Status_ID
Notes: Primary key of a unique and arbitrary value.

2. Open_Status
Notes: Value of 0 or 1, indicating a status of either Open or Closed.

3. Date_Established
Notes: The actual real world date/time that either a particular status or direction or both, began.

4. Date_Retired
Notes: The acutal real world date/time that either a particular status or direction or both ended.

5. Direction
Notes: The direction of the normal flow of traffic along the Transport Link. There are only three values based on the implied direction of the From-To value in the Transport Link this particular status represents: -1 (opposite direction as the From-To), 0 (bidirectional), and 1 (same direction as the From-To).

6. Transport_Link_ID (FK)
Notes: Foreign key to relate back to entity type Transport Link.

Foreign key details (child)
Transport_Link_Link_Status_FK

Definition:	**Child**	**Parent**
	Transport_Link_ID	Transport_Link.Transport_Link_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

LRS_Component

Notes: The object (data, methods, procedures, policies, etc) of a subsystem which by itself is manageable and usable (e.g., the route table, the link/node tables, etc).

Primary key: LRS_Component_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Component_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Interface_Format_Filename	VARCHAR2(255)	Not allowed	
Metadata_Filename	VARCHAR2(255)	Not allowed	

Column details

1. LRS_Component_ID
Notes: Primary key of a unique and arbitrary value.

2. Name
Notes: The name of the component (e.g.,network, route, datum).

3. Interface_Format_Filename
Notes: The information necessary for others (LRS subsystems or LRS apps dev or LRS users) to successfully interface with the subsystem: data formats, policies on use, etc.

4. Metadata_Filename
Notes: The name of the file that contains a myriad of component-specific metadata.

LRS_Component_Xref

Notes: The relationship between a LRS Contact and a given Component.

Primary key: LRS_Component_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Component_Xref_ID	NUMBER(11,0)	Not allowed	
LRS_Contact_ID (FK)	NUMBER(11,0)	Not allowed	
LRS_Component_ID (FK)	NUMBER(11,0)	Allowed	
Type	VARCHAR2(30)	Not allowed	
Description	VARCHAR2(255)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	

Column details

1. LRS_Component_Xref_ID
Notes: Primary key of a unique and arbitrary value.

2. LRS_Contact_ID (FK)
Notes: Foreign Key relating to entity LRS Contact that must be in the same format.

3. LRS_Component_ID (FK)

Notes: Foreign Key relating to entity LRS Component that must be in the same format.

4. Type
Notes: There are three types of relationships:
1) LRS Customer: One who applies the component in some business function or activity.
2) LRS Subsystem Manager: One who is responsible for satisfying the customer and business performance expectations of the component.
3) LRS Subsystem Support Person: One who performs some role in the maintenance and management of the component.

5. Description
Notes: A text string that describes any specific information about the relationship that must be managed over time. For LRS customer, what they use the component for, how often, etc.

6. Date_Established
Notes: When the relationship began.

7. Date_Retired
Notes: When the relationship no longer existed.

Foreign key details (child)
LRS_Comp_LRS_Comp_Xref_FK

Definition:	**Child**	**Parent**
	LRS_Component_ID	LRS_Component.LRS_Component_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

LRS_Contact_LRS_Comp_Xef_FK

Definition:	**Child**	**Parent**
	LRS_Contact_ID	LRS_Contact.LRS_Contact_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

LRS_Contact

Notes: An Iowa DOT staff person who is somehow involved with the LRS.

Primary key: LRS_Contact_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Contact_ID	NUMBER(11,0)	Not allowed	
Person_Name	VARCHAR2(255)	Not allowed	
Organization	VARCHAR2(255)	Not allowed	

Title	VARCHAR2(255)	Not allowed
Voice Phone	VARCHAR2(20)	Not allowed
Fax Phone	VARCHAR2(20)	Allowed
Email Address	VARCHAR2(255)	Allowed
Mailing Address	VARCHAR2(255)	Allowed
Date_Established	DATE	Not allowed
Date_Retired	DATE	Allowed

Column details

1. LRS_Contact_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Person_Name
Notes: Full name.

3. Organization
Notes: What organization the person represents (organization, division, section, unit).

4. Title
Notes: The person's formal title in the organization.

5. Voice Phone
Notes: the telephone number to contact the person.

6. Fax Phone
Notes: The telephone number to contact the person by fax

7. Email Address
Notes: The email address of the contact person.

8. Mailing Address
Notes: The complete mailing address of the contact person.

9. Date_Established
Notes: The date that data is being added.

10. Date_Retired
Notes: The date that data is being removed from active viewing.

LRS_Metadata

Notes: As we anticipate that there will be more than one Iowa DOT referencing system in the future, system-wide metadata is essential. It defines the referencing parameters for the entire system as well as provide vital information to external users of data as well as data providers.

Primary key: LRS_Metadata_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Metadata_ID	NUMBER(11,0)	Not allowed	
Horizontal_Datum	VARCHAR2(255)	Not allowed	
Vertical_Datum	VARCHAR2(255)	Not allowed	
Coordinate_System	VARCHAR2(255)	Not allowed	
Coordinate_System_Units	VARCHAR2(30)	Not allowed	

Coordinate_Format_Filename	VARCHAR2(255)	Not allowed
Datum_Distance_Units	VARCHAR2(30)	Not allowed

Column details

1. LRS_Metadata_ID

Notes: A primary key that is unique and arbitrary. This column is not necessary to distinguish between different Metadata records, since only one is anticipated. Visio, however will not generate a DDL without it.

2. Horizontal_Datum

Notes: Type of referencing mapping surface that provides positional information with reference to a mathematical model of the earth's surface. For instance, the North American Datum of 1983 with a 1991 adjustment (NAD83(91)) with reference to the Clarke spheroid of 1866.

3. Vertical_Datum

Notes: A reference surface that defines the elevation of mapped features on the earth such as the National Geodetic Vertical Datum of 1929 (NGVD29). this datum is based on measurements of mean sea level of 21 tidal stations in the U.S.

4. Coordinate_System

Notes: The type of geographic referencing system being used for the system such as Cartesian, PLSS, etc.

5. Coordinate_System_Units

Notes: What the coordinate system is measured in such as meters, kilometers, etc.

6. Coordinate_Format_Filename

Notes: A text file describing the digital format for coordinate values.

7. Datum_Distance_Units

Notes: Measurement units (metric, English, etc.) of any distance on the datum.

LRS_Subsystem

Notes: The LRS Subsystems.

Primary key: LRS_Subsystem_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Subsystem_ID	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Full_Name	VARCHAR2(255)	Not allowed	
Description	VARCHAR2(255)	Not allowed	

Column details

1. LRS_Subsystem_ID

Notes: A primary key that is unique and arbitrary.

2. Date_Established

Notes: The date/time when the LRS Board of Directors approved the creation of this subsystem.

3. Date_Retired
Notes: The date/time when the LRS Board of Directors approved the elimination of this Subsystem.

4. Full_Name
Notes: The full name of the subsystem, completely spelled out. The current valid entries are: 'Datum Management', 'Route Management', 'Coordinate Route Management', 'Segmental Management', 'Literal Description Management', 'Reference Post Management', 'Milepoint Management', 'Stationing Management'

5. Description
Notes: The subsystem purpose relative to the overall LRS, content from both data and behavior perspective.

LRS_Tables

Notes: The metadata directly related to the individual LRS tables and the individual rows in the LRS tables.

Primary key: LRS_Tables_ID

Columns	Data type	Allow NULLs	Value/Range
LRS_Tables_ID	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Full_Name	VARCHAR2(255)	Not allowed	
LRS_Subsystem_ID (FK)	NUMBER(11,0)	Not allowed	
LRS_Component_ID (FK)	NUMBER(11,0)	Not allowed	
Metadata_Filename	VARCHAR2(255)	Not allowed	

Column details

1. LRS_Tables_ID
Notes: A primary key that is unique and arbitrary.

2. Date_Established
Notes: The official date when the table was initially made ready for access by applications.

3. Date_Retired
Notes: The official date when the table was made no long accessible for read or write purposes.

4. Full_Name
Notes: The full system name of this table, which includes the schema and table name.

5. LRS_Subsystem_ID (FK)
Notes: The foreign key to the table of the same name.

6. LRS_Component_ID (FK)
Notes: The foreign key to the table of the same name.

7. Metadata_Filename
Notes: The name of the file that contains the table level and row level metadata for

this table.

Foreign key details (child)

LRS_Component_LRS_Tables_FK

Definition:	**Child**	**Parent**
	LRS_Component_ID	LRS_Component.LRS_Component_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

LRS_Subsystem_LRS_Tables_FK1

Definition:	**Child**	**Parent**
	LRS_Subsystem_ID	LRS_Subsystem.LRS_Subsystem_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Measurement_Methods

Notes: The method and related technologies used to acquire LRS data. This information provides the range in which data qualities are possible.

Primary key: Measurement_Methods_ID

Columns	Data type	Allow NULLs	Value/Range
Measurement_Methods_ID	NUMBER(6,0)	Not allowed	
Type	VARCHAR2(255)	Not allowed	
Equipment	VARCHAR2(255)	Not allowed	
Procedure_File_Name	VARCHAR2(255)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Change_Reason	VARCHAR2(255)	Allowed	

Column details

1. Measurement_Methods_ID

Notes: Primary key that is a unique arbitrary number for each record in the entity.

2. Type

Notes: The LRS object to which this method of data collection is applied. For example, the 'Datum Anchor Section' would be one type to which several methods may exist.

3. Equipment

Notes: The original measurement tools used to gather the data source (handheld GPS, digitizer).

4. Procedure_File_Name

Notes: A text file to hold a brief description of how the measurement was collected

and processed.

5. Date_Established
Notes: The date when this method was accepted as a valid DOT approach to measurement.

6. Date_Retired
Notes: The date when this method was no longer accepted as a valid DOT approach to measurement.

7. Change_Reason
Notes: A text string that is updated, describing why a method was established and retired.

Nested_Network_Xref

Notes: The relationship between a group of links and nodes that comprise a link or node.

Primary key: Nested_Network_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
Nested_Network_Xref_ID	NUMBER(11,0)	Not allowed	
Element_Type	CHAR(1)	Not allowed	
Element_ID	NUMBER(11,0)	Not allowed	
Aggregated_Element_Type	CHAR(1)	Not allowed	
Aggregated_Element_ID	NUMBER(11,0)	Not allowed	
Aggregated_Element_Name_ID (FK)	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Nested_Network_Xref_ID
Notes: The primary key that is unique and arbitrary.

2. Element_Type
Notes: The type of the element; valid values are only 'Link' or 'Node'.

3. Element_ID
Notes: A link or node that is a component or part of an aggregated link or node. It is the primary key of a link or node, found in the Transport Link or Transport Node table. It must correspond to the value of Element Type.

4. Aggregated_Element_Type
Notes: The type of the element; valid values are only 'Link' or 'Node'.

5. Aggregated_Element_ID
Notes: A link or node that is the complex object that is composed of other links and nodes. It is the primary key of a link or node, found in the Transport Link or Transport Node table. It must correspond to the value of Aggregated Element Type.

6. Aggregated_Element_Name_ID (FK)
Notes: A foreign key that must be exactly the same attribute in the Aggregated Element Name table.

7. Improvement_Project_State_ID
Notes: A foreign key that must be exactly the same attribute in the Improvement Project State table.

Foreign key details (child)
Junction_Nested_Netwrk_Xref_FK

Definition:	**Child**	**Parent**
	Aggregated_Element_Name_ID	Aggregated_Element_Name. Aggregated_Element_Name_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Node_Status

Notes: At a given point in time, the change in condition of whether vehicles can travel through a node.

Primary key: Node_Status_ID

Columns	Data type	Allow NULLs	Value/Range
Node_Status_ID	NUMBER(11,0)	Not allowed	
Open_Status	NUMBER(1,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Transport_Node_ID (FK)	NUMBER(11,0)	Not allowed	

Column details
1. Node_Status_ID
Notes: Primary key of a unique and arbitrary value.

2. Open_Status
Notes: Value of 0 or 1, indicating a status of either Open or Closed.

3. Date_Established
Notes: The actual real world date/time a particular traffic status began.

4. Date_Retired
Notes: The actual real world date/time a particular traffic status ended.

5. Transport_Node_ID (FK)
Notes: Foreign key to relate back to entity type Transport Node

Foreign key details (child)
Transport_Node_Node_Status_FK

Definition:	**Child**	**Parent**
	Transport_Node_ID	Transport_Node.Transport_Node_ID

Cardinality: One -to- Zero-or-More

Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Project_Section

Notes: A section of roadway that is part of the improvement project.

Primary key: Project_Section_ID

Columns	Data type	Allow NULLs	Value/Range
Project_Section_ID	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Allowed	
Project_Section_Node_ID_From (FK)	NUMBER(11,0)	Not allowed	
Project_Section_Node_ID_To (FK)	NUMBER(11,0)	Not allowed	
Station_From	NUMBER(11,0)	Allowed	
Station_To	NUMBER(11,0)	Allowed	
Distance	NUMBER(10,3)	Not allowed	

Column details

1. Project_Section_ID
Notes: Primary key that is a unique arbitrary number for each record in the entity.

2. Improvement_Project_State_ID
Notes: Foreign key to relate to entity type Improvement Project State.

3. Project_Section_Node_ID_From (FK)
Notes: A foreign key to the Project Section Node table.

4. Project_Section_Node_ID_To (FK)
Notes: A foreign key to the Project Section Node table.

5. Station_From
Notes: The stationing value at the beginning of the section. The station value may be negative if it extends before the beginning of the project.

6. Station_To
Notes: The stationing value at the end of the section. The station value may be greater than the project stationing extent if it necessary to place the end of the project section beyond the extent of the project.

7. Distance
Notes: DERIVED - It is calculated from the Station From and Station To values.

Foreign key details (child)
Project_Sec_Node_ID_From_FK

Definition:	**Child**	**Parent**
	Project_Section_Node_ID_From	Project_Section_Node. Project_Section_Node_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action

Ref. Integrity on delete: No Action

Project_Sec_Node_ID_To_FK

Definition:	**Child**	**Parent**
	Project_Section_Node_ID_To	Project_Section_Node. Project_Section_Node_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Project_Section_Node

Notes: The shared location between two or more Project Section ends. The nodes define whether there is a gap in the project between sections or whether cross streets are parts of the project.

Primary key: Project_Section_Node_ID

Columns	Data type	Allow NULLs	Value/Range
Project_Section_Node_ID	NUMBER(11,0)	Not allowed	
Node_Description	VARCHAR2(255)	Not allowed	

Column details

1. Project_Section_Node_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Node_Description
Notes: A verbal description of where this terminus is, typically in terms of signed routes, positions to cities and easily identifiable roadway features (e.g., bridges). This value should be DERIVED by using the Literal Description LRM.

Project_Section_Node_Offset

Notes: The measured location of where the node falls within the datum.

Primary key: Project_Section_Node_Offset_ID

Columns	Data type	Allow NULLs	Value/Range
Project_Section_Node_Offset_ID	NUMBER(11,0)	Not allowed	
Project_Section_Node_ID (FK)	NUMBER(11,0)	Not allowed	
Anchor_Section_ID (FK)	NUMBER(11,0)	Allowed	
Anchor_Section_Offset	NUMBER(10,3)	Not allowed	
Date_Retired	DATE	Allowed	
Measurement_Methods_ID	NUMBER(11,0)	Not allowed	
Measurement_File_Name	VARCHAR2(255)	Not allowed	

Column details

1. Project_Section_Node_Offset_ID
Notes: Primary key of a unique and arbitrary number.

2. Project_Section_Node_ID (FK)
Notes: Foreign key to relate back to entity type Project Section Node.

3. Anchor_Section_ID (FK)
Notes: Foreign key to relate back to entity type Anchor Section.

4. Anchor_Section_Offset
Notes: DERIVED. The measurement of the node on the anchor section offset from an anchor point.

5. Date_Retired
Notes: Derived: either from the Anchor Section or Improvement Project State for the Project Section to which the Project Section Node is related. The reason it is either is because either one can cause the retirement.

6. Measurement_Methods_ID
Notes: Foreign key to relate back to entity type Measurement Methods.

7. Measurement_File_Name
Notes: A text string that points to an external file or database that contains the originally collected information.

Foreign key details (child)

Proj_Sec_Nd_Proj_Sec_Nd_Off_FK

Definition:	**Child**	**Parent**
	Project_Section_Node_ID	Project_Section_Node. Project_Section_Node_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	Cascade	
Ref. Integrity on delete:	Cascade	

Anch_Sec_Proj_Sec_Nd_Off_FK

Definition:	**Child**	**Parent**
	Anchor_Section_ID	Anchor_Section.Anchor_Section_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Ref_Feature_Route_Xref

Notes: The mapping of a reference feature with a route that traverses a Transport Link along which the feature exists.

Primary key: Ref_Feature_Route_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
Ref_Feature_Route_Xref_ID	NUMBER(11,0)	Not allowed	
Reference_Feature_ID (FK)	NUMBER(11,0)	Not allowed	
Route_ID	NUMBER(11,0)	Not allowed	

Column details

1. Ref_Feature_Route_Xref_ID
Notes: The primary key that is unique and arbitrary.

2. Reference_Feature_ID (FK)
Notes: A foreign key that is the same attribute as found in the Reference Feature table.

3. Route_ID
Notes: A foreign key of the same attribute found in the Route table.

Foreign key details (child)
Ref_Feat_Ref_Feat_Rt_Xref_FK

Definition:	**Child**	**Parent**
	Reference_Feature_ID	Reference_Feature.Reference_Feature_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Reference_Feature

Notes: A physical object in the field along a transportation facility. Its purpose is to provide a known and consistent point from which to locate objects and events that occur along the facility.

Primary key: Reference_Feature_ID

Columns	Data type	Allow NULLs	Value/Range
Reference_Feature_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Transport_Link_Offset	NUMBER(10,3)	Not allowed	
Transport_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Reference_Feature_Category_ID (FK)	NUMBER(6,0)	Allowed	
Source_ID	VARCHAR2(30)	Allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Measurement_Methods_ID	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Reference_Feature_ID

Notes: Primary key that is a unique and arbitrary number.

2. Name
Notes: DERIVED: A text string that names the feature, pulled from the source data base.

3. Transport_Link_Offset
Notes: CALCULATED: Based on the offset distance from the beginning of the anchor section.

4. Transport_Link_ID (FK)
Notes: Foreign key to relate back to entity Transport Link.
CALCULATED: Based on the relationship between anchor section and the transport link.

5. Reference_Feature_Category_ID (FK)
Notes: Foreign key to relate back to entity type Reference Feature Category. There are only two categories: Bridge or railroad crossing reference features.

6. Source_ID
Notes: The actual primary key of the feature object from its source data base (e.g., the unique and arbitrary number for a bridge span).

7. Date_Established
Notes: The date the feature was added as a reference feature.

8. Date_Retired
Notes: The date the feature can no longer be used as a reference feature.

9. Measurement_Methods_ID
Notes: A foreign key (relating to the entity type Measurement Methods) that associates the method used to collect the Anchor Section Offset. The associated method provides the quality of the offset and whether it node's position meets reference object accuracy standards.

10. Improvement_Project_State_ID
Notes: Foreign key to relate back to entity type Improvement Project State

Foreign key details (child)

Trans_Link_Ref_Feature_FK

	Child	**Parent**
Definition:	Transport_Link_ID	Transport_Link.Transport_Link_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Ref_Feat_Category_Ref_Feat_FK

	Child	**Parent**
Definition:	Reference_Feature_Category_ID	Reference_Feature_Category. Reference_Feature_Category_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed

Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Reference_Feature_Category

Notes: The DOT accepted reference features and the specific piece of the refererence feature which is used as the point of reference.

Primary key: Reference_Feature_Category_ID

Columns	Data type	Allow NULLs	Value/Range
Reference_Feature_Category_ID	NUMBER(6,0)	Not allowed	
Type	VARCHAR2(20)	Not allowed	
Location_Desc	VARCHAR2(255)	Not allowed	

Column details

1. Reference_Feature_Category_ID
Notes: The primary key that is unique and arbitrary.

2. Type
Notes: The reference feature object itself. Currently, there are only two support reference features: bridges and railroad crossings.

3. Location_Desc
Notes: The specific component or element of the reference feature type that is used as the point of reference (e.g., an expansion joint on reference feature type bridge).

Reference_Post

Notes: Posts placed at approximately 1-mile increments along a signed route. Formerly known as a milepost.

Primary key: Reference_Post_ID

Columns	Data type	Allow NULLs	Value/Range
Reference_Post_ID	NUMBER(11,0)	Not allowed	
Route_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Route_ID	NUMBER(11,0)	Not allowed	
Post_Value	NUMBER(5,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Transport_Link_ID	NUMBER(11,0)	Not allowed	
Transport_Link_Offset	NUMBER(10,3)	Not allowed	
Anchor_Section_ID	NUMBER(11,0)	Allowed	
Anchor_Section_Offset	NUMBER(10,3)	Allowed	
Full_Name	VARCHAR2(255)	Allowed	
County_ID	NUMBER(4,0)	Allowed	
GIMS_MilePoint_Value	NUMBER(8,0)	Allowed	

LAT	NUMBER(8,5)	Allowed
LNG	NUMBER(8,5)	Allowed
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed

Column details

1. Reference_Post_ID
Notes: Primary key that is a unique and arbitrary number.

2. Route_Link_ID (FK)
Notes: Foreign key that must be exactly the same attribute as in the Route Link.

3. Route_ID
Notes: DERIVED: Foreign key that must be exactly the same attribute as in the Route.

4. Post_Value
Notes: The value that is posted (e.g., '289').

5. Date_Established
Notes: The date that the reference post is established in the field with the marker value. Dates that reflect when the post was destroyed (purposely or by accident) are managed as part of the business system that maintains the markers.

6. Date_Retired
Notes: The date that the reference post no longer portrays the marker value. Dates that reflect when the post was destroyed (purposely or by accident) are managed as part of the business system that maintains the markers.

7. Transport_Link_ID
Notes: DERIVED: Foreign key that must be exactly the same attribute as in the Transport Link.

8. Transport_Link_Offset
Notes: CALCULATED: A distance measurement along the link where the reference post is positioned. Because of the accuracy requirements of LRS reference objects, the Reference Post distances (d1) must be measured in the field as an offset from the beginning of an anchor section. Transport Nodes will also be reference objects and will have the same requirement (d2). Therefore, the Link Offset of the Reference Post meets the same accuracy requirements (d1-d2 = offset value). The Anchor Section distance for the Reference Post is not stored as data (since it is just data used in the process). The Link Offset value is because this format meets the LRM data requirements.

9. Anchor_Section_ID
Notes: Foreign key relating to entity type Anchor Section.

10. Anchor_Section_Offset
Notes: The distance that this Reference Post exists from the From Anchor Point of the Anchor Section.

11. Full_Name
Notes: Define Attribute from the Reference Post legacy system.

12. County_ID
Notes: Define Attribute from the Reference Post legacy system.

13. GIMS_MilePoint_Value
Notes: Define Attribute from the Reference Post legacy system.

14. LAT

Notes: Define Attribute from the Reference Post legacy system.

15. LNG
Notes: Define Attribute from the Reference Post legacy system.

16. Improvement_Project_State_ID
Notes: Foreign key relating to entity type Improvement Project State.

Foreign key details (child)
Route_Link_Reference_Post_FK

Definition:	**Child**	**Parent**
	Route_Link_ID	Route_Link.Route_Link_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route

Notes: The master list of assigned routes over the LRS networks. It is primarily used to be the placeholder of the primary key unique across all routes, and is part of the LRS. The individual route systems are not used within the LRS business logic.

Primary key: Route_ID

Columns	Data type	Allow NULLs	Value/Range
Route_ID	NUMBER(11,0)	Not allowed	
Full_Name	VARCHAR2(255)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Route_System_ID (FK)	NUMBER(11,0)	Not allowed	

Column details
1. Route_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Full_Name
Notes: DERIVED: Hold the full name for a particular route concatenated from information in the route systems through the Assigned Routes table. Concatenation rules vary within and between route systems.

3. Date_Established
Notes: DERIVED: The date/time when the route was officially made a route by the governing body of the system (legislature, city council, DOT, etc). This is the same value found in the Assigned_Routes table for this particular Route

4. Date_Retired
Notes: DERIVED: The date/time when the route was officially retired by the governing body of the system (legislature, city council, DOT, etc). This is the same value found in the Assigned_Routes table for this particular Route

5. Route_System_ID (FK)
Notes: Foreign key to relate back to entity Route System.

Foreign key details (child)
Route_System_Route_FK

	Child	Parent
Definition:	Route_System_ID	Route_System.Route_System_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_Alias

Notes: Official alias names for LRS routes.

Primary key: Route_Alias_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Alias_ID	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Route_Jurisdiction_ID (FK)	NUMBER(11,0)	Not allowed	
Full_Name	VARCHAR2(255)	Not allowed	
Route_ID (FK)	NUMBER(11,0)	Not allowed	

Column details

1. Route_Alias_ID
Notes: A primary key that is unique and arbitrary.

2. Date_Established
Notes: The date when the alias was made official by the jurisdiction organization.

3. Date_Retired
Notes: The date when the alias was officialy retired by the jurisdiction organization.

4. Route_Jurisdiction_ID (FK)
Notes: A foreign key that must be exactly the same attribute as in the Route Jurisdiction table.

5. Full_Name
Notes: The full name of the Alias completely spelled out. The name includes all route name components: geographic extent, prefix, name, type, and suffix. These are not separate in this table because Alias names do not require any of these fields.

6. Route_ID (FK)
Notes: Foreign key relating to table Route.

Foreign key details (child)
Route_Route_Alias_FK

	Child	Parent
Definition:	Route_ID	Route.Route_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Route_Juris_Route_Alias_FK

	Child	Parent
Definition:	Route_Jurisdiction_ID	Route_Jurisdiction.Route_Jurisdiction_ID
Cardinality:	One -to- One-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Route_Jurisdiction

Notes: The ultimate organization that establishes and retires a route name assignment to a roadway.

Primary key: Route_Jurisdiction_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Jurisdiction_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Common_Code	VARCHAR2(20)	Allowed	

Column details

1. Route_Jurisdiction_ID

Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Name

Notes: The full name of the organization. For example, "Iowa State University - Ames Campus Board of Regents".

3. Common_Code

Notes: The DOT code if one is already assigned to this organization.

Route_Link

Notes: The sequence of links that compose a route. A route-link is the actual traversal of a Iowa DOT route along the network: an ordered and directed,

but not necessarily connected, set of whole links. Conventions are required for establishing traversal directionality (in contrast to link directionality) and for specifying non-connected traversals (from NCHRP 20-27(2).

Primary key: Route_Link_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Link_ID	NUMBER(11,0)	Not allowed	
Ordinal	NUMBER(5,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Distance	NUMBER(10,3)	Not allowed	
Transport_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Route_ID (FK)	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Route_Link_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Ordinal
Notes: The ordinal value (e.g., 1,2,3, etc) that defines the sequence of a link in a route. This sequence is based on the Federal definition of sequencing south to north and west to east. This is the only place that defines route topology along the network.

3. Date_Established
Notes: Date/time when a route was approved to traverse along a link or set of links (e.g., a detour, a jurisdictional transfer, etc).

4. Date_Retired
Notes: Date/time when a route no longer traverses a link or set of links (e.g., the end of a detour, a jurisdictional transfer, etc).

5. Distance
Notes: DERIVED: The datum distance derived from Transport_Link for the length of the link.

6. Transport_Link_ID (FK)
Notes: Foreign key relating to table Transport Link.

7. Route_ID (FK)
Notes: Foreign key relating to table Route.

8. Improvement_Project_State_ID
Notes: Foreign key relating to table Improvement Project State.

Foreign key details (child)

Transport_Link_Route_Link_FK

Definition:	**Child**	**Parent**
	Transport_Link_ID	Transport_Link.Transport_Link_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_Route_Link_FK

Definition:	**Child** Route_ID	**Parent** Route.Route_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Route_Prefix

Notes: A qualifer to a route. For most posted routes, this is a directional classification.

Primary key: Route_Prefix_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Prefix_ID	NUMBER(6,0)	Not allowed	
Name	VARCHAR2(20)	Not allowed	
DOT_Abbreviation	VARCHAR2(6)	Allowed	

Column details

1. Route_Prefix_ID

Notes: A primary key that is unique and arbitrary.

2. Name

Notes: The full spelled out name in its entirety (e.g., 'North' or 'West').

3. DOT_Abbreviation

Notes: The offiical DOT abbreviation for this suffix. DOT wants one and only one official abbreviation. For example, 'N' for 'North'.

Route_Prefix_Alias

Notes: The enumerated list of potential aliases for a given prefix that most likely will be encounted when data is submitted or provided to DOT. For example, for North', entries here may be 'N', 'Nrth', 'North, 'north', 'nort', etc.

Primary key: Route_Prefix_Alias_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Prefix_Alias_ID	NUMBER(6,0)	Not allowed	
Alias	VARCHAR2(6)	Not allowed	
Route_Prefix_ID (FK)	NUMBER(6,0)	Allowed	

Column details

1. Route_Prefix_Alias_ID

Notes: A primary key that is unique and arbitrary.

2. Alias
Notes: The actual alias value.

3. Route_Prefix_ID (FK)
Notes: A foreign key to the Route_Prefix table. An alias can belong to one and only one Route Prefix.

Foreign key details (child)
Rt_Prefix_Rt_Prefix_Alias_FK

Definition:	**Child**	**Parent**
	Route_Prefix_ID	Route_Prefix.Route_Prefix_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_Suffix

Notes: A qualifer to a route. For state routes, this is a classification (mainline, business, etc). For streets, it is typicaly a qualifier of direction.

Primary key: Route_Suffix_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Suffix_ID	NUMBER(6,0)	Not allowed	
Name	VARCHAR2(20)	Not allowed	
DOT_Abbreviation	VARCHAR2(6)	Allowed	

Column details
1. Route_Suffix_ID
Notes: A primary key that is unique and arbitrary.

2. Name
Notes: The spelled out full name of the suffix. For example, for state routes, 'Mainline'. For streets, 'North West'.

3. DOT_Abbreviation
Notes: The offiical DOT abbreviation for this suffix. DOT wants one and only one official abbreviation. For example, 'Mn' for 'Mainline', and 'NW' for North West' .

Route_Suffix_Alias

Notes: The enumerated list of potential aliases for a given suffix that most likely will be encounted when data is submitted or provided to DOT. For example, for 'Mainline', entries here may be 'MN', 'Mn', 'mn', 'Ml', 'Mainline', etc.

Primary key: Route_Suffix_Alias_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Suffix_Alias_ID	NUMBER(6,0)	Not allowed	
Alias	VARCHAR2(6)	Not allowed	
Route_Suffix_ID (FK)	NUMBER(6,0)	Not allowed	

Column details

1. Route_Suffix_Alias_ID
Notes: A primary key that is unique and arbitrary.

2. Alias
Notes: The actual alias value.

3. Route_Suffix_ID (FK)
Notes: A foreign key to the Route_Suffix table. An alias can belong to one and only one Route Suffix.

Foreign key details (child)
Rt_Suffix_Rt_Suffix_Alias_FK

Definition:	**Child**	**Parent**
	Route_Suffix_ID	Route_Suffix.Route_Suffix_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_System

Notes: The enumerated list of formal route systems managed by the DOT and its customers.

Primary key: Route_System_ID

Columns	Data type	Allow NULLs	Value/Range
Route_System_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Schema	VARCHAR2(30)	Not allowed	

Column details

1. Route_System_ID

2. Name
Notes: The spelled out name of the route system. Some examples are 'State of Iowa Signed Routes', 'Institutional Signed Routes,' 'Emergency 911 Signed

Routes', 'Private Routes', etc.

3. Schema
Notes: The computer system name of the schema that contains the route system. All route systems have the same tabular contents.

Route_Type

Notes: A classification of state routes (interstate, US, or Iowa) or streets (Street, Boulevard, Avenue, etc).

Primary key: Route_Type_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Type_ID	NUMBER(6,0)	Not allowed	
Name	VARCHAR2(20)	Not allowed	
DOT_Abbreviation	VARCHAR2(6)	Allowed	

Column details

1. Route_Type_ID
Notes: A primary key that is unique and arbitrary.

2. Name
Notes: The full spelled out name in its entirety (e.g., 'Street' or 'Interstate').

3. DOT_Abbreviation
Notes: The offiical DOT abbreviation for this type. DOT wants one and only one official abbreviation. For example, 'St' for 'Street', 'I' for Interstate, and 'US' for United States'.

Route_Type_Alias

Notes: The enumerated list of potential aliases for a given type that most likely will be encounted when data is submitted or provide to DOT. For example, for 'Street', entries here may be 'St', 'ST', 'ST.', 'Str', 'Street', etc.

Primary key: Route_Type_Alias_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Type_Alias_ID	NUMBER(6,0)	Not allowed	
Alias	VARCHAR2(6)	Not allowed	
Route_Type_ID (FK)	NUMBER(6,0)	Not allowed	

Column details

1. Route_Type_Alias_ID
Notes: A primary key that is unique and arbitrary.

2. Alias
Notes: The actual alias value.

3. Route_Type_ID (FK)
Notes: A foreign key to the Route_Type table. An alias can belong to one and only one Route Type.

Foreign key details (child)
Route_Type_Route_Type_Alias_FK

Definition:	**Child**	**Parent**
	Route_Type_ID	Route_Type.Route_Type_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_Type_Alias_Ramp

Notes: The enumerated list of potential aliases for a given type that most likely will be encounted when data is submitted or provide to DOT. For example, for 'Ramp', entries here may be 'Rmp', 'Ramp', etc.

Primary key: Route_Type_Alias_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Type_Alias_ID	NUMBER(6,0)	Not allowed	
Alias	VARCHAR2(6)	Not allowed	
Route_Type_ID (FK)	NUMBER(6,0)	Not allowed	

Column details
1. Route_Type_Alias_ID
Notes: A primary key that is unique and arbitrary.

2. Alias
Notes: The actual alias value.

3. Route_Type_ID (FK)
Notes: A foreign key to the Route_Type table. An alias can belong to one and only one Route Type.

Foreign key details (child)
Rt_Typ$Rmp_Rt_Typ_Alias$Rmp_FK

Definition:	**Child**	**Parent**
	Route_Type_ID	Route_Type_Ramp.Route_Type_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Route_Type_Ramp

Notes: A classification of ramps. Valid values are "Ramp" and "Connector".

Primary key: Route_Type_ID

Columns	Data type	Allow NULLs	Value/Range
Route_Type_ID	NUMBER(6,0)	Not allowed	
Name	VARCHAR2(20)	Not allowed	
DOT_Abbreviation	VARCHAR2(6)	Allowed	

Column details

1. Route_Type_ID

Notes: A primary key that is unique and arbitrary.

2. Name

Notes: The full spelled out name in its entirety (e.g., 'Street' or 'Interstate').

3. DOT_Abbreviation

Notes: The offiical DOT abbreviation for this type. DOT wants one and only one official abbreviation. For example, 'St' for 'Street', 'I' for Interstate, and 'US' for United States'.

Station_Marker

Notes: Markers placed at approximately 500' increments along a roadway based on an improvement project.

Primary key: Station_Marker_ID

Columns	Data type	Allow NULLs	Value/Range
Station_Marker_ID	NUMBER(11,0)	Not allowed	
Station	VARCHAR2(15)	Not allowed	
Station_Marker_Type_ID (FK)	NUMBER(6,0)	Allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Project_Section_Offset	NUMBER(10,3)	Not allowed	
Project_Section_ID (FK)	NUMBER(11,0)	Not allowed	

Column details

1. Station_Marker_ID

Notes: Primary key that is a unique and arbitrary number.

2. Station

Notes: The value that is posted as the station.

3. Station_Marker_Type_ID (FK)

Notes: Foreign key to relate to entity type Station Marker Type.

4. Date_Established
Notes: The date that the marker is established in the field with the marker value. Dates that reflect when the post was destroyed (purposely or by accident) are managed as part of the business system that maintains the markers.

5. Date_Retired
Notes: The date that the marker no longer portrays the marker value. Dates that reflect when the post was destroyed (purposely or by accident) are managed as part of the business system that maintains the markers.

6. Project_Section_Offset
Notes: A distance measurement along the Project Section where the Station Post is positioned.

7. Project_Section_ID (FK)
Notes: Foreign key that must be exactly the same attribute as in the Project Section.

Foreign key details (child)
Proj_Section_Station_Post_FK

	Child	Parent
Definition:	Project_Section_ID	Project_Section.Project_Section_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Stat_Post_Type_Stat_Post_FK

	Child	Parent
Definition:	Station_Marker_Type_ID ype_ID	Station_Marker_Type.Station_Marker_T
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Station_Marker_Type

Notes: Enumerated entity type to describe the different methods in which a station marker is placed in the field.

Primary key: Station_Marker_Type_ID

Columns	Data type	Allow NULLs	Value/Range
Station_Marker_Type_ID	NUMBER(6,0)	Not allowed	
Type	VARCHAR2(30)	Not allowed	

Column details

1. Station_Marker_Type_ID

Notes: Primary key that is a unique and arbitrary number.

2. Type

Notes: There are only two types of markers: A 'Station Post' and a 'Station Pavement Stamp".

Subproject_Phase

Notes: The phases that an improvement project moves through. For roadway improvement projects, examples are planning, design, and construction.

Primary key: Subproject_Phase_ID

Columns	Data type	Allow NULLs	Value/Range
Subproject_Phase_ID	NUMBER(6,0)	Not allowed	
Phase	VARCHAR2(255)	Not allowed	
Description	VARCHAR2(255)	Not allowed	

Column details

1. Subproject_Phase_ID

Notes: The primary key that is unque and arbitrary.

2. Phase

Notes: The name fo the phase spelled out completely. Valid phases for roadway improvement projects are 'Roadway Improvement Conceptual', 'Roadway Improvement Strategic Planning', 'Roadway Improvement Programming', 'Roadway Improvement Project Planning, 'Roadway Improvement Preliminary Engineering', 'Roadway Improvement Design', 'Roadway Improvement As-let', and 'Roadway Improvement As-Built'.

3. Description

Notes: A general description about the purpose of the phase, the scope of the phase (first and last tasks), and what product/output from the phase.

Subproject_Type

Notes: Enumerated entity type to describe the different Project/subproject Types: alignment, bridge, environmental, utility, survey, ROW, etc.

Primary key: Subproject_Type_ID

Columns	Data type	Allow NULLs	Value/Range
Subproject_Type_ID	NUMBER(6,0)	Not allowed	
Type	VARCHAR2(255)	Not allowed	
Description	VARCHAR2(255)	Not allowed	

Column details

1. Subproject_Type_ID
Notes: The primary key that is unquie and arbitrary.

2. Type
Notes: The type of subproject it is. For roadway improvement projects, some example types are 'Roadway Improvement Alignment', 'Roadway Improvement Bridge', 'Roadway Improvement Environmental', 'Roadway Improvement Utility, 'Roadway Improvement Survey', 'Roadway Improvement Right of Way', etc.

3. Description
Notes: A general description about what the type is (and is not), general characteristics.

Transport_Link

Notes: In general, the transport link is a "A topological connection between two ordered nodes." (NCHRP 20-27(2). At Iowa DOT, the Transport Link represents the ability to travel from one transport node to another.

Primary key: Transport_Link_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Link_ID	NUMBER(11,0)	Not allowed	
Transport_Node_ID_From (FK)	NUMBER(11,0)	Not allowed	
Transport_Node_ID_To (FK)	NUMBER(11,0)	Not allowed	
Distance	NUMBER(11,3)	Not allowed	
Open_Status	NUMBER(1,0)	Not allowed	
Direction	NUMBER(1,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_Link_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Transport_Node_ID_From (FK)
Notes: The node where the link begins. The ID is gathered from the Transport Node entity.

3. Transport_Node_ID_To (FK)
Notes: The node where the link ends. The ID is gathered from the Transport Node entity.

4. Distance
Notes: DERIVED: The datum distance derived from Anchor Section for the length of the link.

5. Open_Status
Notes: The curent condition of the link, based on the values in the Link Status Table. This attribute has the same defintion and values as the Link Status attribute.

6. Direction

Notes: The current direction of the transport link, based on the values in the Link Status Table. This attribute is the same in definition and value as the Link Status attribute.

7. Date_Established
Notes: The real world date when the link is first opened for traffic.

8. Date_Retired
Notes: The date when the link is permanently closed to traffic.

9. Improvement_Project_State_ID
Notes: Foreign key to relate back to entity type Improvement Project State.

Foreign key details (child)
Transport_Node_ID_From_FK

Definition:	**Child**	**Parent**
	Transport_Node_ID_From	Transport_Node.Transport_Node_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_Node_ID_To_FK

Definition:	**Child**	**Parent**
	Transport_Node_ID_To	Transport_Node.Transport_Node_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_Link_AS_Xref

Notes: A derived table from information in the Transport Link and Anchor Section conflation object. Needed only for LRSx, entirely derived.

Primary key: Transport_Link_AS_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Link_AS_Xref_ID	NUMBER(11,0)	Not allowed	
From_Position	NUMBER(10,3)	Not allowed	
To_Position	NUMBER(10,3)	Not allowed	
Transport_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Anchor_Section_ID (FK)	NUMBER(11,0)	Allowed	

Column details
1. Transport_Link_AS_Xref_ID

2. From_Position
Notes: The proportion (0..100) of where a segment of a Transport Link begins along

anchor section.

3. To_Position
Notes: The proportion (0..100) of where a segment of a Transport Link ends along anchor section.

4. Transport_Link_ID (FK)
Notes: A foreign key that has the same definition of the attribute found in the Transport Link Table.

5. Anchor_Section_ID (FK)
Notes: A foreign key that has the same definition of the attribute found in the Anchor Section Table.

Foreign key details (child)

Trans_Lnk_Trans_Lnk_AS_Xref_FK

Definition:	**Child**	**Parent**
	Transport_Link_ID	Transport_Link.Transport_Link_ID

Cardinality: One -to- One-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Anch_Sec_Trans_Link_AS_Xref_FK

Definition:	**Child**	**Parent**
	Anchor_Section_ID	Anchor_Section.Anchor_Section_ID

Cardinality: One -to- One-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_Mode

Notes: Enumerated entity type to describe the means of transportation - roadway, railway, pedestrian, bike, intermodal, etc.

Primary key: Transport_Mode_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Mode_ID	NUMBER(6,0)	Not allowed	
Transport_Mode	VARCHAR2(20)	Not allowed	

Column details

1. Transport_Mode_ID
Notes: Primary key that is a unique and arbitrary number.

2. Transport_Mode
Notes: The mode name, fully spelled out.

Transport_Node

Notes: "A zero-dimensional object that is a topological junction of two or more links, or end point of a link. They are located geometrically by reference to the datum." This entity includes the information that comprises the Literal Description LRM for intersection features.

Primary key: Transport_Node_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Node_ID	NUMBER(11,0)	Not allowed	
Open_Status	NUMBER(1,0)	Not allowed	
Reference_Feature	NUMBER(1,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_Node_ID
Notes: Primary key that is a unique arbitrary for each record in the entity.

2. Open_Status
Notes: The current status of the node - this is the most current value from the Node Status. This attribute is the same definition and values as the Node Status attribute.

3. Reference_Feature
Notes: This is a Boolean attribute ('Yes' or 'No') to indicate whether the node is designated as a reference feature as part of the Literal Description linear reference method.

4. Date_Established
Notes: The real world date when the link is first opened for traffic.

5. Date_Retired
Notes: The date when the link is permanently closed to traffic.

6. Improvement_Project_State_ID
Notes: Foreign key to relate back to entity type Improvement Project State

Transport_Node_Offset

Notes: An intersection table relates the many-to-many relationships between the anchor section and the transport node.

Primary key: Transport_Node_Offset_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Node_Offset_ID	NUMBER(11,0)	Not allowed	

Transport_Node_ID (FK)	NUMBER(11,0)	Not allowed
Anchor_Section_ID (FK)	NUMBER(11,0)	Allowed
Anchor_Section_Offset	NUMBER(10,3)	Not allowed
Date_Retired	DATE	Allowed
Measurement_Methods_ID (FK)	NUMBER(11,0)	Not allowed
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed

Column details

1. Transport_Node_Offset_ID
Notes: Primary key to be a unique and arbitrary number.

2. Transport_Node_ID (FK)
Notes: Foreign key to relate to entity type Transport Node.

3. Anchor_Section_ID (FK)
Notes: Foreign key to relate to entity type Anchor Section.

4. Anchor_Section_Offset
Notes: This is the datum measure attribute that is used to locate the node on the anchor section. It is an offset measure from the "from" anchor point of the anchor section and is expressed as a distance measure in the same units as the "distance" attribute of the associated anchor section.

5. Date_Retired
Notes: Derived from either the Anchor Section or Transport Node date retired; whichever caused the retirement.

6. Measurement_Methods_ID (FK)
Notes: A foreign key (relating to the Measurement Methods entity type) that associates the method used to collect the Anchor Section Offset. The associated method provides the quality of the offset and whether it node's position meets reference object accuracy standards.

7. Improvement_Project_State_ID
Notes: Foreign key to relate to entity type Improvement Project State.

Foreign key details (child)

Trans_Node_Trans_Node_Off_FK

Definition:	**Child**	**Parent**
	Transport_Node_ID	Transport_Node.Transport_Node_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	Cascade	
Ref. Integrity on delete:	Cascade	

Msrmt_Meth_Trans_Node_Off_FK

Definition:	**Child**	**Parent**
	Measurement_Methods_ID	Measurement_Methods. Measurement_Methods_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Anch_Sec_Trans_Node_Off_F

Definition:	**Child**	**Parent**

	Anchor_Section_ID	Anchor_Section.Anchor_Section_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_Node_Route_Xref

Notes: All On Route - At Route Signed Route combinations found at a particular node in the network. For example, US 30E as the at route, and Main Street as the On Route, and vice versa, would be two rows in this table.

Primary key: Transport_Node_Route_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_Node_Route_Xref_ID	NUMBER(11,0)	Not allowed	
Transport_Node_ID (FK)	NUMBER(11,0)	Not allowed	
On_Route_ID	NUMBER(11,0)	Not allowed	
At_Route_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_Node_Route_Xref_ID
Notes: The primary key that is unique and arbitrary.

2. Transport_Node_ID (FK)
Notes: A foreign key that represents the same attribute in the Transport Node table.

3. On_Route_ID
Notes: The route on which the feature to be referenced exists.

4. At_Route_ID
Notes: The cross route from which the feature location is offset.

Foreign key details (child)

Trans_Nd_Trans_Nd_Rt_Xref_FK

Definition:	Child	Parent
	Transport_Node_ID	Transport_Node.Transport_Node_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_System

Notes: A set of system route links serving a common purpose or are representing

transport facilities that are commonly managed. Some key examples are the primary, secondary, municipal, and parks/institutions systems. Other examples are transit systems, snowplow systems, and so on. Transport Systems are typically networks, but do not have to be networks. That is, a system can be non-contiguous sections of transport facilities, like road sections.

Primary key: Transport_System_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_System_ID	NUMBER(11,0)	Not allowed	
Name	VARCHAR2(255)	Not allowed	
Type	VARCHAR2(255)	Not allowed	
Extent	VARCHAR2(255)	Not allowed	
Transport_Mode_ID (FK)	NUMBER(6,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_System_ID
Notes: Primary key that is a unique and arbitrary number for each Transport System.

2. Name
Notes: A text field for naming the system. (e.g., City of Ames Public Road System).

3. Type
Notes: Combinations of routes that serve a purpose and run over a combination of route categories (e.g., municipal network).

4. Extent
Notes: The geographic region that the particular transport system covers such as statewide, city of Ames, etc.

5. Transport_Mode_ID (FK)
Notes: Foreign key to relate to entity type Transport Mode.

6. Improvement_Project_State_ID
Notes: Foreign key to relate to entity type Improvement Project State.

Foreign key details (child)
Trans_Mode_Trans_System_FK

Definition:	**Child**	**Parent**
	Transport_Mode_ID	Transport_Mode.Transport_Mode_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Transport_System_Milepoint

Notes: The official LRS milepoint values for only the route-links of a given route that are within the selected Transport System.

Primary key: Transport_System_Milepoint_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_System_Milepoint_ID	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Accumulated_Begin_Distance	NUMBER(10,3)	Not allowed	
Accumulated_End_Distance	NUMBER(10,3)	Not allowed	
Route_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Transport_System_ID (FK)	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_System_Milepoint_ID
Notes: The primary key that is unique and arbitrary.

2. Date_Established
Notes: The date this particular version of the milepoint route was created.

3. Date_Retired
Notes: The date this particular version of the milepoint route was made obsolete.

4. Accumulated_Begin_Distance
Notes: For a given link along a route within a transport system, the derived accumulated distance along a route at the beginning of the link. The begin distance is based on the ordinal nature of the route and not on the Transport Link From-To. As a result, the Transport Link From may have a higher accumlated distance value than a Transport Link To.

5. Accumulated_End_Distance
Notes: For a given link along a route within a transport system, the derived accumulated distance along a route at the end of the link. The end distance is based on the ordinal nature of the route and not on the Transport Link From-To. As a result, the Transport Link From Node may have a higher accumlated distance value than the To Node on the same Link.

6. Route_Link_ID (FK)
Notes: A foreign key that is the same attribute found in the Route Link table.

7. Transport_System_ID (FK)
Notes: A foreign key that is the same attribute found in the Transport System table.

8. Improvement_Project_State_ID
Notes: A foreign key that is the same attribute found in the Improvement Project State table.

Foreign key details (child)

Rt_Link_Trans_Sys_Milepoint_FK

Definition:	**Child**	**Parent**
	Route_Link_ID	Route_Link.Route_Link_ID

Cardinality: One -to- Zero-or-More
Allow NULLs: Not allowed
Ref. Integrity on update: No Action
Ref. Integrity on delete: No Action

Trans_Sys_Trans_Sys_Milept_FK

Definition:	**Child**	**Parent**

	Transport_System_ID	Transport_System.Transport_System_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

Transport_System_Xref

Notes: The association of a Transport System and the links that compose the Transport System.

Primary key: Transport_System_Xref_ID

Columns	Data type	Allow NULLs	Value/Range
Transport_System_Xref_ID	NUMBER(11,0)	Not allowed	
Date_Established	DATE	Not allowed	
Date_Retired	DATE	Allowed	
Transport_System_ID (FK)	NUMBER(11,0)	Not allowed	
Transport_Link_ID (FK)	NUMBER(11,0)	Not allowed	
Improvement_Project_State_ID	NUMBER(11,0)	Not allowed	

Column details

1. Transport_System_Xref_ID
Notes: The primary key that is unqiue and arbitrary.

2. Date_Established
Notes: The official date when a link was assigned to the transport system.

3. Date_Retired
Notes: The official date when a link was removed from a particular transport system.

4. Transport_System_ID (FK)
Notes: A foreign key that is the same attribute as found in the Transport System table.

5. Transport_Link_ID (FK)
Notes: A foreign key that is the same attribute found in the Transport Link table.

6. Improvement_Project_State_ID
Notes: A foreign key that is the same attribute found in the Improvement Project State table.

Foreign key details (child)
Trans_Link_Trans_Sys_Xref_FK

Definition:	**Child**	**Parent**
	Transport_Link_ID	Transport_Link.Transport_Link_ID
Cardinality:	One -to- Zero-or-More	

Allow NULLs:	Not allowed
Ref. Integrity on update:	No Action
Ref. Integrity on delete:	No Action

Trans_Sys_Trans_Sys_Xref_FK

Definition:	**Child**	**Parent**
	Transport_System_ID	Transport_System.Transport_System_ID
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Ref. Integrity on update:	No Action	
Ref. Integrity on delete:	No Action	

3.3 Tables and Columns of Extended Data Types

ANCHOR_POINT

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_DATUM	MDSYS.SDO_GEOMETRY	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_DATUM

Notes: A Spatial object datatype that stores the geometric representation of a Datum feature.

2. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of an anchor point automatically derived from GIMS cartography using the Centerline's relationship with Anchor Sections.

ANCHOR_SECTION

Columns	Data type	Allow NULLs	Value/Range
CONFLATION_GIMSCARTO	TDSYS.LRSX_CONFLATION	Not allowed	
GEOMETRY_DATUM	MDSYS.SDO_GEOMETRY	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1 CONFLATION_GISMCARTO

Notes: An array of LRSX_SEGMENT types that define a one-to-many mapping of a row in a LRM table to rows in a table containing cartographic features. The conflation object stores the name of the table and the name of the column in the table that the geometry for this column should be derived from.

2. GEOMETRY_DATUM

Notes: A Spatial object datatype that stores the geometric representation of a Datum feature.

3. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of an anchor section automatically derived from GIMS cartography using the Centerline's relationship with Anchor Sections.

CENTERLINE

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a cartographic feature based on GIMS cartography.

GIMS_CONTROL_SECTION

Columns	Data type	Allow NULLs	Value/Range
CONFLATION_ANCHOR_SECTIO	TDSYS.LRSX_CONFLATION	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1 CONFLATION_ANCHOR_SECTION

Notes: An array of LRSX_SEGMENT types that define a one-to-many mapping of a row in a LRM table to rows in a table containing cartographic features. The conflation object stores the name of the table and the name of the column in the table that the geometry for this column should be derived from.

HPMS_CONTROL_SECTION

Columns	Data type	Allow NULLs	Value/Range
CONFLATION_ANCHOR_SECTIO	TDSYS.LRSX_CONFLATION	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1 CONFLATION_ANCHOR_SECTION

Notes: An array of LRSX_SEGMENT types that define a one-to-many mapping of a row in a LRM table to rows in a table containing cartographic features. The conflation object stores the name of the table and the name of the column in the table that the geometry for this column should be derived from.

2. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a control section automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

PROJECT_SECTION

Columns	Data type	Allow NULLs	Value/Range
CONFLATION_ANCHOR_SECTION	TDSYS.LRSX_CONFLATION	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1 CONFLATION_ANCHOR_SECTION

Notes: An array of LRSX_SEGMENT types that define a one-to-many mapping of a row in a LRM table to rows in a table containing cartographic features. The conflation object stores the name of the table and the name of the column in the table that the geometry for this column should be derived from.

2. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a project section, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

PROJECT_SECTION_NODE

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a project node, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

REFERENCE_POST

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a reference post, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

STATION_MARKER

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a station marker, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

TRANSPORT_LINK

Columns	Data type	Allow NULL	Value/Rnge
CONFLATION_ANCHOR_SECTION	TDSYS.LRSX_CONFLATION	Not allowed	
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1 CONFLATION_ANCHOR_SECTION

Notes: An array of LRSX_SEGMENT types that define a one-to-many mapping of a row in a LRM table to rows in a table containing cartographic features. The conflation object stores the name of the table and the name of the column in the table that the geometry for this column should be derived from.

2. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a transport link, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

TRANSPORT_NODE

Columns	Data type	Allow NULLs	Value/Range
GEOMETRY_GIMSCARTO	MDSYS.SDO_GEOMETRY	Not allowed	

Column details

1. GEOMETRY_GIMSCARTO

Notes: A Spatial object datatype that stores the geometric representation of a transport node, automatically derived from GIMS cartography though the Centerline's conflation relationship with Anchor Sections.

4 APPENDIX A – DATA STRUCTURE AND FORMAT STANDARDS

1. Table and Column Name Capitalization

- GeoMedia requires all Oracle tables and columns to be in capital letters.
- If table names are single case, Oracle and other tools can use any upper/lower case variation of the name and it will still find the correct table and column. Names that are actually assigned mixed case labels must be referred to by applications with the name in quotes, causing complexity and confusion.
- Visio allows mixed case entry of table names, which helps for readability. When the DDL is generated, however, all table and column names are converted to upper case.

2. Table Naming

- Table names will be spelled out in full. For example, 'ANCHOR_POINT', not 'ANCHOR_PT' or 'AP'
- Cross-reference table names or tables that share common entities in the name shall use the full names of the entities. The table purpose, indicating what brought these two entities together can be abbreviated. The table name, however, must be less than 28 characters long (see Column Naming; Primary Keys). If this is not possible, the entity names can then be abbreviated but should use a common abbreviation for that entity. For example, 'ANCHOR_POINT_CENTERLINE_CTL', not 'ANCHOR_POINT_CNTRLNE_CONTROL' or 'AP_CENTERLINE_CONTROL'

3. Column Naming

- Primary Keys: must be a combination of the table name with the suffix "_ID".
- Other than the primary key, do not include the table name in full or in abbreviation in a column name.
- Using succinct column names results in a cleaner database structure by eliminating the presence of redundant characters and the need to abbreviate names.
- This may require the use of the table name in queries to distinguish between columns with the same name. In Oracle queries, the table name can be assigned a shorter alias, which is then used to refer to the table. Being forced to use the table name when referring to a column makes for a more explicit syntax in the code, and therefore clearer.

4. Common Columns

- Foreign Keys: The names of these keys are always spelled out completely, even if the value in the column is filled by a process / application (rather than enforced by the database).
- Dates: 'DATE_ESTABLISHED' and 'DATE_RETIRED' will be used throughout for date columns that refer to analogous events.
- Offsets: <entity_name>_OFFSET will be used for linear offset values. For example, ANCHOR_SECTION_OFFSET in the REFERENCE_POST table.

5. Enumeration Handling

Except for special circumstance, all enumerations will be stored as a look up table in Oracle, and will be referred to in other tables by a foreign key. The name of the look up table should refer to the element being enumerated. The primary key value could be character if that is more appropriate.

6. State and Status Terminology

The term ***state*** means an object can exist in two or more conditions – a state column is enumerated. The term ***status*** means an object is in one or another condition – a status column is Boolean. Boolean values are not valid Oracle datatypes, however, so they will be represented in the database by "0" if the value is True and "1" if the value is False.

7. Oracle Schema Naming

For the Pilot, we will have at least two schemas: LLRSPILOT (linear location reference system) and GDWPILOT (geodata warehouse).