OmniTRANS

Transport Planning Software

OmniTRANS

Database Description

Version 6.0

Basic Information

Document Description

This document describes the database structure of OmniTRANS.

Document Version

Number: 1.00
Date: 13 April 2011

OmniTRANS

Version: 6.0

If you are using a different version of OmniTRANS then it is possible that the information provided in this document may differ from what you see in your version of the program.

How to Contact Omnitrans International

	Snipperlingsdijk 4 7417 BJ Deventer The Netherlands	Address
☎	+31-(0)570-666111 +31-(0)570-666888	Phone Fax
✉	P.O. Box 161 7400 AD Deventer The Netherlands	Mail
	info@omnitrans.nl	Email
	www.omnitrans-international.com www.omnitrans.nl	Website
@	techsupport@omnitrans.nl	Technical support

Contents

1 Introduction

This document describes the database structure of OmniTRANS Version 6.0[1]. The OmniTRANS database is a relational database, currently implemented in Paradox. The data is divided over many different tables in order to reduce redundancy and to make the storage of data efficient and flexible. For performance reasons, referential integrity is maintained by the OmniTRANS application and not the database itself.

Chapter 2 explains important concepts used in OmniTRANS. Chapter 3 describes the structure of the standard database, table by table.

Conventions in this document
When the structure of a table is given, this always starts with the name of the table in capitals and then between brackets the fieldnames. When the fieldname is a key field (index), the name is in *italic*.

[1] Since OmniTRANS Version 5.1, the user can extend the database and add custom fields for a selected set of objects. This document describes the default system fields.

2 Database Structure

2.1 Projects and Variants

The data of an OmniTRANS project is divided over different subdirectories within a project. All the data that is general for the project resides in the project directory. All the data that is relevant for a variant resides in a directory with the name of the variant. This data comprises both input data and output data. A subvariant is a subdirectory under the variant directory. A subvariant inherits the input data from the variant and has only output data stored in the subdirectory. “Project general data” is referred to as level 0 data, “variant input data” as level 1 data and “variant output data” as level 2 data.

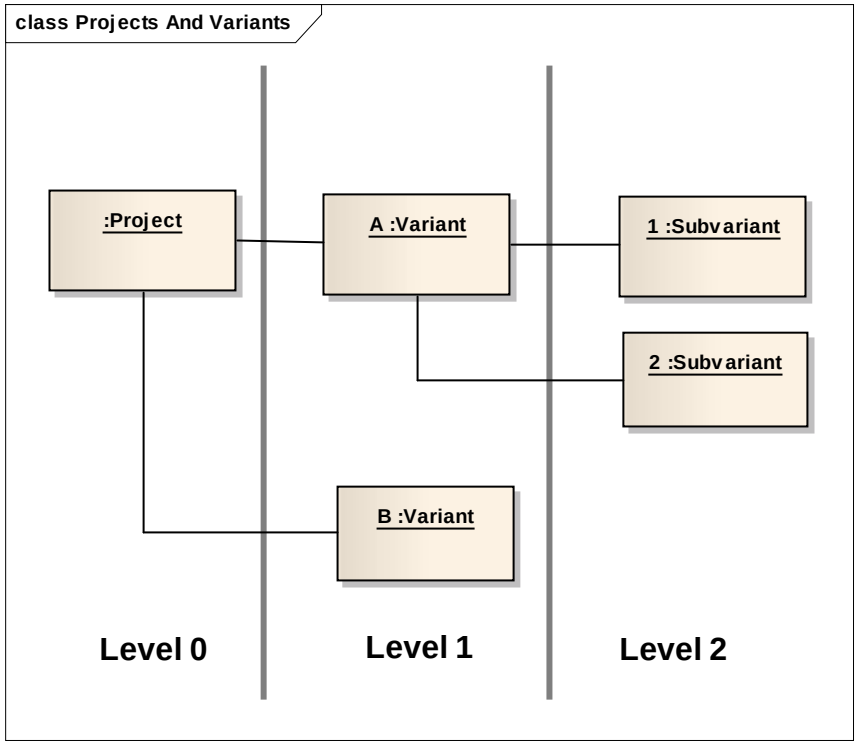

Figure 1: Project, Variant and Subvariant structure

Typical examples of level 0 data are the graphical representations of objects, such as the coordinates of a node, the project setup data, such as the dimensions, and the matrix cube data. The matrices themselves are not stored in the database but in binary files in the matrix directory within the project.
Examples of level 1 data are the presence of an object in this variant, the attributes of an object, such as speed and capacity for links or junction data for nodes.
Typical examples of level 2 data are the loads on links or the skim-matrices.

2.2 OmniTRANS Objects

The OmniTRANS database is defined around objects. Examples of an object can be a node, a link or a turn. Each object can have data at different levels. Its graphical representation will be stored in level 0 tables, its attributes will be stored in level 1 tables and its generated output data will be stored in level 2 tables. Within each level the data could be divided over different tables.

OmniTRANS version 6.0 knows the following objects:

- Area
- Centroid (zone)
- Node
- Link
- Turn
- Count
- Screenline
- Stop
- Transitline
- Controls[2]
- Routes (User-defined) [3]
- Matrices

The graphical representations of all the objects, stored in level 0 tables, together form the super-network.

2.3 Table Naming Convention

OmniTRANS uses the following naming conventions for the data tables associated with an object (italic parameters are primary key fields).

- OBJECT(*key*)
- OBJECT1DATA1(*key*,d1..dn)
- OBJECT2DATA1(*key,serienr*, typenr)
- OBJECT3DATA1(*key,mode,time*, d1..dn)
- OBJECT4DATA1(*key,purpose,mode,time,user*, d1..dn)
- OBJECT5DATA1(*key,purpose,mode,time,user,result,iteration*, d1..dn)
- OBJECT6DATA1(*key, combinationnr*, d1..dn)
- OBJECT7DATA1(*key, combinationnr,result,iteration*,d1..dn)
- OBJECT8DATA1(*key, datasetnr,*,d1..dn)

So an object is defined with a table where the name of the table is the name of the object and the key of the object (unique id) is the name of the object followed by nr. Data related

[2] Controls are available within specific versions of OmniTRANS.
[3] Schematic routes are available specific versions of OmniTRANS.

directly to this object can be stored in OBJECT1DATA1. When more data is related directly to the object you can either add a field to OBJECT1DATA1 or have a second table called OBJECT1DATA2. The type information of the object resides in the table Object2Data1. Data related to the object but different per mode and time is stored in Object3Data1. Data related to the object but different per purpose, mode, time and user is stored in OBJECT4DATA1, etc..

Example for a node:

- NODE(*nodenr*)
- NODE1DATA1(*nodenr*, junctiontype, ralanes, coordfactor)
- NODE1DATA2(*nodenr*, nodetag, zoneboundary)
- NODE2DATA1(*nodenr, serienr*, typenr)
- NODE5DATA1(*nodenr*, *purpose, mode, time, user, result, iteration*, vcratio, wvcratio, delay, backofqueue, los, calccycletime, conflictratio, usedsignaltype, lossvehhours)

For data related to an object that need extra keys, the convention is that the number of extra keys is inserted in between the object name, the type of data and the word data. So this second digit (m) denotes the number of extra keys:

- OBJECT1_MDATA1..N (*key*,extrakey1..m, d1..dn)
- OBJECT2_MDATA1..N (*key,serienr, extrakey1..m*, typenr)
- OBJECT3_MDATA1..N (*key,mode,time, extrakey1..m*, d1..dn)
- OBJECT4_MDATA1..N (*key,purpose,mode,time,user,extrakey1..m*, d1..dn)
- OBJECT5_MDATA1..N (*key,purpose,mode,time,user,result,iteration,extrakey1..m*,d1..dn)
- OBJECT6_MDATA1..N (*key,combinationnr,extrakey1..m*,d1..dn)
- OBJECT7_MDATA1..N (*key,combinationnr,result,iteration,extrakey1..m*,d1..dn)
- OBJECT8_MDATA1..N (*key,datasetnr,extrakey1..m*,d1..dn)

Example for link which have as extra key the direction:

- LINK1_1DATA1(*linknr*, *direction*, lanesmask, widthcr, exitlanes, etc..)

So this table has one extra key field: direction.

When two objects are related to each other, such as a count to a link, or a stop to node, OmniTRANS store the relation in a table that uses the following naming conventions:

- OBJECTA2OBJECTB (*keya,* keyb)

Example of a count to a link:

- COUNT2LINK(*countnr,* linknr, direction)

Note that in this example, the primary key of the table COUNT2LINK is *countnr* and that linknr and direction are used to refer to the associated link and direction.

2.4 Super Network

The super network is constructed of basic graphical objects: points, lines, trilines, polylines and polygons. All objects in OmniTRANS can be classified according to these basic drawing objects. For example centroids and nodes are points, links are lines, turns are trilines, screenlines and transit lines are polylines and areas are polygons.

A distinction between different objects is made using a type field.

The following basic object tables exist.

1 POINT(*pointnr*, *pointtype*, x, y, namenr)
2 LINE(*linenr*, *linetype*, pointnrA, pointtypeA, pointnrB, pointtypeB, namenr,shape)
3 TRILINE(*trilinenr*, *trilinetype*, pointnrA, pointtypeA, pointnrB, pointtypeB, pointnrC, pointtypeC, namenr)
4 POLYLINE(*polylinenr*, *polylinetype*, pointype, namenr)
POLYLINEPOINTS(*polylinenr*, *polylinetype*, *ordernr*, pointnr)
5 POLYGON(*polygonnr*, *polygontype*, namenr, shape)

Polyline objects are stored using two graphical tables. The table POLYLINE stores the polyline objects and the table POLYLINEPOINTS stores the points composing each polyline with their order. A line and a polygon can be composed of multiple points for defining their contour (at least 3 points for a polygon) and this information is stored in the field shape in the form of binary large object.

The field *namenr* points to a record in the table with all the object names:

- NAME(*object, objecttype*, *namenr*, name)

In this table object refers to the graphical object and object type refers to the different types with a graphical object. For example within polyline there are two object types: screenline (1) and transitline (2).

The relation between an object and the graphical representation is given in the following table.

Graphical object	Object (field)	OmniTRANS Object	Objecttype (field)
Point	1	Centroid	1
Point	1	Node	2
Point	1	Count	4
Point	1	Stop	5
Point	1	Control	7
Line	2	Link	1
Line	2	Segment	2
Triline	3	Turn	1
Polyline	4	Screenline	1
Polyline	4	Transitline	2
Polyline	4	Route (User-defined)	3
Polygon	5	Area	1

Table 1: Relations between graphical objects and OmniTRANS objects

2.5 Links

A link is a transport channel connecting either two nodes or one centroid to a node. A link is defined by a point A and a point B, these points being either centroids or nodes.
The visual representation of a link does not have to be a direct line between these two points. It can be shaped according to the corresponding geographical physical contour. The collection of points and segments making the geographical physical contour of a link are saved as a single entity within a BLOB (Binary Large Object) field. In this way, OmniTRANS allows the user to match a link with its exact contour while keeping the number of links and nodes to a minimum for the performance of the transport algorithms.

A similar concept is used for managing the points and segments associated with a polygon or area.

2.6 Matrix cube and Matrices

Matrices are organised in matrix cubes. The data of a matrix cube follows the same principles as the data for other objects, with the only difference that they don't have a graphical representation. OD-matrices reside at level 0 tables, while other (generated) matrix types, such as skims and selected-links reside at level 2, because they are treated as output data.

Tables for matrixcube

- CUBE(*cubenr*, name,size)
- CUBE4DATA1(*cubenr,purpose,mode,time,user*,filename)

The filename refers to a binary file that resides in the matrix directory.

The data related to a matrixcube, such as zonal data and trip-ends are stored in the following tables:

- CUBE1_1DATA1(*cubenr, centroidnr*, zonal data fields...)
- CUBE4_1DATA1(*cubenr, purpose, mode, time, user, centroidnr*, growthcolumntotal, growthcolumnweight, growthrowtotal, growthrowweight)
- CUBE6_1DATA1(*cubenr, combinationnr, centroidnr*, tripend, tripendweight)

The first table contains the zonal data (socio-economic data) as specified by the user. The second table contains the row and column totals as used for growth factoring. The third table contains the tripends per combination. The field tripend either contains a production or an attraction. This depends on how the combination is specified.

At level 2, in a variant, more matrices are stored using the table:

- CUBE5DATA1(*cubenr, purpose, mode, time, user, result, iteration,* filename)

Per pmturi a filename is stored. The actual matrices are stored in binary files in the same directory as the variant. The binary files have fixed extension. Per variant you can have a fixed number of output matrix cubes which have negative cube numbers:

Matrix type	**cubenr**	**extension**
Skim	-1	skm
Selected link	-2	slm
Cordon	-3	clm
Screenline	-4	scm
Transit	-5	stm

2.7 Types and Type-items and Modes per Link type

Each object can have different types. In OmniTRANS you can define a type with one or more type-items and attach a type to one or more objects.

The information for types and type-items and the relation to an object is stored at level 0 in the following two tables:

- SERIE(*serienr*, name, status, usage)
- TYPE(*serienr, typenr*, name, etc.)

The field usage is a bitmask specifying which objects relate to a certain type.

Note: Serie is the old name for a type in OmniTRANS 3 and previous versions. From version 3.2, it has been renamed type. The database however still uses the old name.

Each object that has a certain type-item in a certain type has a data table that specifies this relation. So for example a node will have a data table with the following fields.

- NODE2DATA1(*nodenr, serienr*, typenr)

The node can relate to more then one type, so more records in these tables are allowed. A node can have only one type-item per type (typenr per serienr).

The field status in the serie table indicates among other which type is the linktype to which the modes are related. For this type an extra data table exists:

- TYPE3DATA1(*typenr, mode*, *time*, speed, capacity, freespeed, satflow, speedatcap)

This table specifies per typenr which modes are allowed to traverse links of this type. Per mode, time combination the default speed, capacity, free speed, saturation flow and speed at full capacity is given. It is possible for this table to have up to five extra fields. These extra fields depend on the attributes defined for a link per mode and time stored in the table LINK3_1DATA1. The field definitions between the tables TYPE3DATA1 and LINK3_1DATA1 always need to remain synchronized. The OmniTRANS Database Designer automatically takes care of it. The number of field definitions has been limited to 10 in these tables to avoid performance degradation.

2.8 Dimensions and Combinations

The structure of dimensions is stored in a table at level 0:

- DIMENSION(*dimensiontype, dimensionnr*, name, status, parent)

In this table the dimensiontype is a value between one and six to indicate respectively if the dimension is a purpose, a mode, a time, a user, a result or an iteration.

The structure of combinations is stored in two tables at level 0:

- COMBINATION(*combinationtype, combinationnr*, name)
- COMBINATION2PMTU(*combinationtype, combinationnr*, *purpose, mode, time, user*)

The first table defines the combination and its name. The combinationtype is a value of one, two or three to indicate respectively if the combination is a production, an attraction or a screenline combination.

The second table specifies the pmtu's per combination.

2.9 Dataset

Each OmniTRANS network object can have one or more datasets associated with it. Datasets are used to store the history of real time information associated with it

The information for datasets is stored at level 0 in the following two tables:

- DATASET(*datasetnr*, parent, name, description, + user-defined fields)
- DATASETRECORD(*datasetrecordnr, datasetnr*, + user-defined fields)

Each object type that has datasets associated with it has a data table that specifies this relation. So for example a control will have a data table with the following fields.

- CONTROL8DATA1(*controlnr, datasetnr*, + user-defined fields)

2.10 Selections

OmniTRANS has the capability to store selections of objects. The selections are stored in two tables:

- SELECTION(*selectionnr, object, objecttype,* name, shape)
- SELECTIONOBJECTS(*selectionnr, objectnr, direction*)

The first table contains the selection number, the type of objects that the selection applies to and the shape of the polygon associated with the selection if applicable. The type is determined by the object (point,line,triline,polyline or polygon) and the sub-type within that graphical object (objecttype). For a list of graphical objects and associated objecttypes, refer to the table in Section 2.4. The second table stores the objects composing a selection.

When a job run is started the current selection in the network is always written to this table with selection number zero.

2.11 Data dictionary

The database structure is read from a file that resides in the project (level 0). This file has a binary structure and is called *datadictionary.ddy*. The GUI of OmniTRANS reads this file and shapes the interface according to the object found and the data related to these objects.

3 The Standard Database

The database is divided into two sections. The first section defines all the level 0 tables while the second section describes all the level 1 and 2 tables. Field names in *italic* are primary key fields.

3.1 Level 0 tables

Level 0 tables are located in the root of the project. These tables define the project setup, the matrix cubes, the super network and stored selections.

3.1.1 Project Setup tables

The following tables define the project setup:

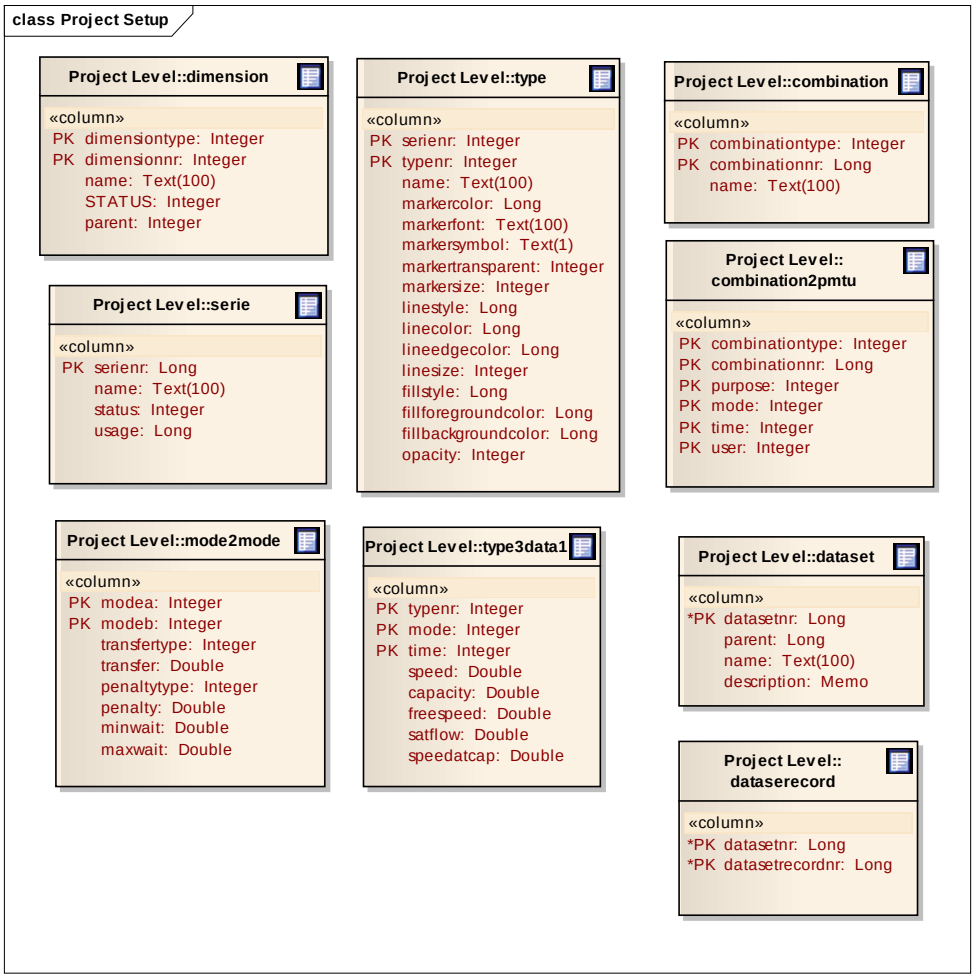

Figure 1: Project Setup Tables

Table Name: DIMENSION

This table stores the dimension setup. It contains the number and names of the dimensions and the hierarchical structure.

Field Name	Field type	Explanation
dimensiontype	Short	Type of the dimension, range 1 to 6. Respectively purpose, mode, time, user, result, iteration
dimensionnr	Short	Number of the dimension
name	String(100)	Name of the dimension
status	Short	Status of this dimension: This is a bitmask: 1st bit: Marked as network 2nd bit: Marked as transit 3rd bit: Marked as walk
parent	Short	Reference to the parent dimensionnr. Zero if the dimension is in the root.

Table Name: SERIE

This table together with the table TYPE defines all the types and type-items. This table stores the possible types and to which objects these types can be associated to.

Field Name	Field type	Explanation
serienr	Short	Number of the type. (serie is the old name for a type)
name	String(100)	Name of the type
status	Short	Status of the type. This is a bitmask: 1st bit: Invisible type 2nd bit: Read only type 3rd bit: Link type
usage	Integer	This field is a bitmask and indicates with which objects the series is associated with: 1st bit: Centroids 2nd bit: Nodes 4th bit: Counts 5th bit: Stops 8th bit: Links 10th bit: Turns 11th bit: Screenlines 12th bit: Transit lines 13th bit: Areas 14th bit: Controls 15th bit: Routes

Table Name: TYPE

This table together with the previous table SERIE defines all the types and type-items. This table stores the display characteristics of each type item.

Field Name	Field type	Explanation
serienr	Short	Number of the type. (serie is the old name for a type)
typenr	Short	Number of the type-item
name	String(100)	Name of the type-item
markercolor	Integer	Colour of the marker (when the type applies to a point object)
markerpicture	Graphic	Picture associated with the marker
markerfont	String(100)	Indicates the font associated with the marker. It is not yet in use.
markersymbol	Char(1)	Indicates the symbol associated with the marker
markertransparent	Short	Toggle controlling the transparency of the marker
markersize	Short	Size of the marker
linestyle	Integer	Style of the line such as dot, solid, dash, arrows, etc..
linecolor	Integer	Colour of the line (when the type applies to a line object)
lineedgecolor	Integer	Colour of the outline of the line
linesize	Short	Size or thickness of the line
fillstyle	Integer	Pattern associated with a marker
fillforegroundcolor	Integer	Foreground colour associated with the type item
fillbackgroundcolor	Integer	Background colour associated with the type item
opacity	Short	Opacity associated with type object. It can vary between 0% and 100%.

Table Name: TYPE3DATA1

This table defines the modes allowed per link type. It also contains the default link attributes per mode and time.

Field Name	Field type	Explanation
typenr	Short	Number of the link type-item.
mode	Short	Number of the mode
time	Short	Number of the time
speed	Float	Default speed
capacity	Float	Default capacity
freespeed	Float	Default speed at free flow
satflow	Float	Default saturation flow
speedatcap	Float	Default speed at capacity

The fields in this table should match the fields in LINK3_1DATA1. They are the default values.

Table Name: COMBINATION

This table stores the combinations and their names. It is closely related to the table COMBINATION2PMTU which stores the valid pmtu's associated with a combination.

Field Name	Field type	Explanation
combinationtype	Short	Type of the combination: 1: Productions 2: Attractions 3: Screenlines
combinationnr	Short	Number of the combination
name	String(100)	Name of the combination

Table Name: COMBINATION2PMTU

This table stores the valid pmtu's associated with a combination.

Field Name	Field type	Explanation
combinationtype	Short	Type of the combination: 1: Productions 2: Attractions 3: Screenlines
combinationnr	Short	Number of the combination
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension

Table Name: MODE2MODE

This table is only used for transit. It contains attributes for the transfer between two transit modes.

Field Name	Field type	Explanation
modea	Short	Number of the transit mode from
modeb	Short	Number of the transit mode to
transfertype	Short	Type for the next field: 1: Factor 2: Constant
transfer	Float	Transfer factor or constant when transferring from modea to modeb
penaltytype	Short	Type for the next field: 1: Factor

		2: Constant
penalty	Float	Penalty factor or constant when transferring from modea to modeb
minwait	Float	Minimum wait time when transferring from modea to modeb
maxwait	Float	Maximum wait time when transferring from modea to modeb

Table Name: DATASET

This table stores the list of datasets in the project

Field Name	Field type	Explanation
datasetnr	Integer	Dataset number
parent	Integer	Dataset set number associated with the parent dataset if applicable
name	String(100)	Name of the data set
description	Memo	Textual description of the data set

Table Name: DATASETRECORD

This table stores the data records composing a dataset.

Field Name	Field type	Explanation
datasetnr	Integer	Dataset number
datasetrecordnr	Integer	Number of the data record associated with a dataset

3.1.2 Super Network tables

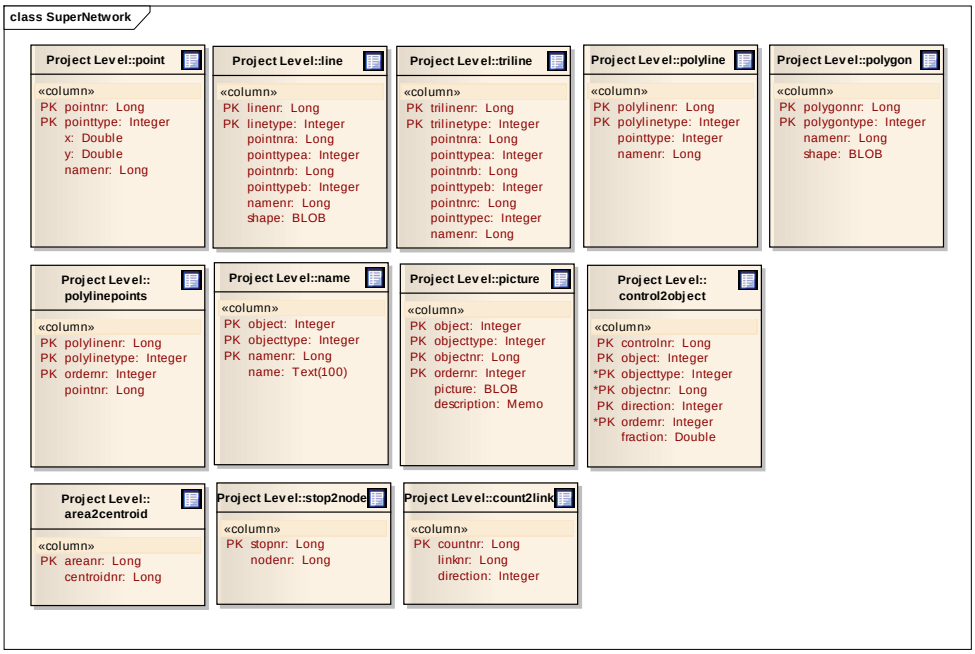

Figure 2: Super Network Tables

Table Name: POINT

This table stores all the point type objects (centroids, nodes, shapepoints, counts, stops, area-points) with their coordinates and name.

Field Name	Field type	Explanation
pointnr	Integer	Number of the point
pointtype	Short	Type of the point: 1: Centroid 2: Node 4: Counts 5: Stops 7: Controls
x	Float	x-coordinate in the format corresponding to the selected project coordinate system.
y	Float	y-coordinate in the format corresponding to the selected project coordinate system.
namenr	Integer	Reference to the name stored in the table NAME

Note that centroid numbers and node numbers can overlap. So you can have centroid one and node one.

Table Name: LINE

This table stores all the line type objects (links) with their a- en b-points and name.

Field Name	Field type	Explanation
linenr	Integer	Number of the line
linetype	Short	Type of the line: 1: Link
pointnra	Integer	Number of point A of the line
pointtypea	Short	Type of point A: 1: Centroid 2: Node
pointnrb	Integer	Number of point B of the line
pointtypeb	Short	Type of point B: 1: Centroid 2: Node
namenr	Integer	Reference to the name stored in the table NAME
shape	BLOB	Collection of points defining the geometry of the object

The lines are always defined from the lowest point number to the highest point number. A centroid is always lower then a node.

Table Name: TRILINE

This table stores all the triline type objects (turns) with their a-, b- and c-points and name.

Field Name	Field type	Explanation
trilinenr	Integer	Number of the triline
trilinetype	Short	Type of the triline: 1: Turn
pointnra	Integer	Number of point A of the triline
pointtypea	Short	Type of point A: 1: Centroid 2: Node
pointnrb	Integer	Number of point B of the triline
pointtypeb	Short	Type of point B: 1: Centroid 2: Node
pointnrc	Integer	Number of point C of the triline
pointtypec	Short	Type of point C: 1: Centroid 2: Node
namenr	Integer	Reference to the name stored in the table NAME

Table Name: POLYLINE

This table stores all the polyline type objects (screenlines, transit lines) with their name. The actual points of the polyline are stored in POLYLINEPOINTS.

Field Name	Field type	Explanation
polylinenr	Integer	Number of the polyline
polylinetype	Short	Type of the polyline: 1: Screenline 2: Transit line 3: Route (user-defined)
pointtype	Short	Type of the point: 2: Node 4: Counts
namenr	Integer	Reference to the name stored in the table NAME

Screenlines have counts as their points. Transit lines and user-defined routes have nodes as their points.

Table Name: POLYLINEPOINTS

This table stores all the polyline points. This table defines a polyline together with the table POLYLINE.

Field Name	Field type	Explanation
polylinenr	Integer	Number of the polyline
polylinetype	Short	Type of the polyline: 1: Screenline 2: Transit line
ordernr	Integer	Order number of the point in the polyline. Numbering starts with one.
pointnr	Integer	Point number in the polyline

Table Name: POLYGON

This table stores the polygon type objects (areas) with their name. The actual points of the polygon are stored in POLYGONPOINTS.

Field Name	Field type	Explanation
polygonnr	Integer	Number of the polygon
polygontype	Short	Type of the polygon: 1: Area
namenr	Integer	Reference to the name stored in the table NAME
shape	BLOB	Collection of points defining the geometry

Table Name: NAME

This table stores all the names associated with the OmniTRANS objects. Each name has a number which is referenced from the different super network tables.

Field Name	Field type	Explanation
object	Short	Object shape 1: Point 2: Line 3: Triline 4: Polyline 5: Polygon
objecttype	Short	Type of the object per object shape: 1: centroid, link, turn, screenline, area 2: Node, transit line 3: Route (user-defined) 4: Counts 5: Stops 7: Controls
namenr	Integer	Name number
name	String(100)	Name

Table Name: PICTURE

This table stores pictures or drawings associated with the OmniTRANS objects. The field *ordernr* allows multiple pictures to be associated with an OmniTRANS object.

Field Name	Field type	Explanation
object	Short	Object shape 1: Point 2: Line 3: Triline 4: Polyline 5: Polygon
objecttype	Short	Type of the object per object shape: 1: centroid, link, turn, screenline, area 2: Node, transitline 3: Route (user-defined) 4: Counts 5: Stops 7: Controls
objectnr	Integer	Identifier of the object to which the picture is associated

ordernr	Short	Sequence order of the picture
picture	Graphics	BLOB containing the picture
description	Memo	Description of the picture.

Table Name: AREA2CENTROID

This table stores the relations between area and centroid. An area can only have one centroid. Although rarely the case, a centroid can technically be associated with multiple areas.

Field Name	Field type	Explanation
areanr	Integer	Area number
centroidnr	Integer	Centroid number

Table Name: STOP2NODE

This table stores the relations between stop and node. A stop can only be associated with one node.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
nodenr	Integer	Node number

Table Name: COUNT2LINK

This table stores the relations between a count and a link. A count can only be associated with one link in one direction. The OmniTRANS application prevents attaching a link in a direction to multiple counts.

Field Name	Field type	Explanation
countnr	Integer	Count number
linknr	Integer	Link number
direction	Short	Direction of the link

Table Name: CONTROL2OBJECT

This table stores the relation between controls and other network objects. The field *ordernr* allows multiple controls to be associated with an OmniTRANS object.

Field Name	Field type	Explanation
controlnr	Integer	Control number
object	Short	Object shape 1: Point 2: Line 3: Triline 4: Polyline 5: Polygon

objecttype	Short	Type of the object per object shape: 1: centroid, link, turn, screenline, area 2: Node, transitline 3: Routes (user-defined) 4: Counts, 5: Stops 7: Controls
objectnr	Integer	Identifier of the object to which the control is associated
direction	Short	Direction of the object (relevant for links)
ordernr	Integer	Sequence order of the control
fraction	Float	Indicates the fraction ??

3.1.3 Matrix Cubes and Zonal Data tables

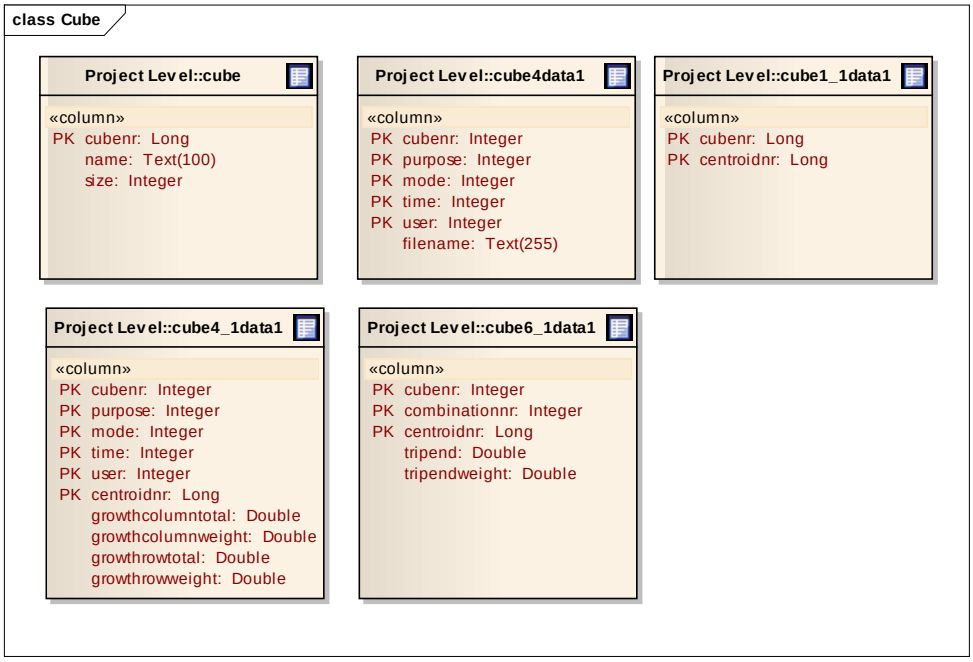

Figure 3: Matrix Cube Tables

At level 0 only the trip matrices are stored. You can have many trip matrices organized per matrix cube.

Table Name: CUBE

This table stores the name of the various matrix cubes available in the project. The matrix cubes provide a structure to manage the various input trip matrices and store different assumptions regarding the traffic demand.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
name	String(100)	Name associated with the cube
size	Short	This field is not used. This field was intended to indicate the number of zones associated with the cube to facilitate the initialization of arrays.

Note that a project has always a cube named @@Scratch@@ that is not visible to the user but is used internally by the OmniTRANS application.

Table Name: CUBE4DATA1

This table stores the filename containing the matrix data for a particular cube, purpose, mode, time and user.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
filename	String(255)	Filename of the binary file with the matrix data

The filename refers to a binary file that resides in the matrix directory. These binary matrices are stored in the directory matrix.

From version 5 onwards, the fare tables associated with the modelling of public transit are also stored using the cube data structure. Cubenr -8 is used for storing fare tables.

The data related to a matrixcube, such as zonal data and trip-ends are stored in the following tables:

Table Name: CUBE1_1DATA1

This table stores the social economic data for each zone (defined by a centroid) and for each matrix cube.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
centroidnr	Integer	Centroid number
...	Float/Int	Arbitrary number of fields with zonal data as defined by the user.

Table Name: CUBE4_1DATA1

This table stores the trip ends which are valid for one matrix (one pmtu). They are typically used for growth factoring. The table is accessible in the growth factoring tab page of the zonal data window.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
centroidnr	Integer	Centroid number

growthcolumntotal	Float	Growth column total (attraction)
growthcolumnweight	Float	Weight for the growth column total
growthrowtotal	Float	Growth row total (production)
growthrowweight	Float	Weight for the growth row total

Table Name: CUBE6_1DATA1

This table contains trip ends per combination. The combination number determines if the trip end is a production or an attraction. This table is used to store trip ends which are valid for multiple matrices (one combination = multiple pmtu). They are typically used for the gravity model. The table is accessible in the combination trip ends tab page of the zonal data window.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
comnbinationnr	Short	Combination number (production or attraction)
centroidnr	Integer	Centroid number
tripend	Float	Production or attraction for the given combination.
tripendweight	Float	Weight of the production or attraction.

3.1.4 Selections

Selections are stored in the following two tables.

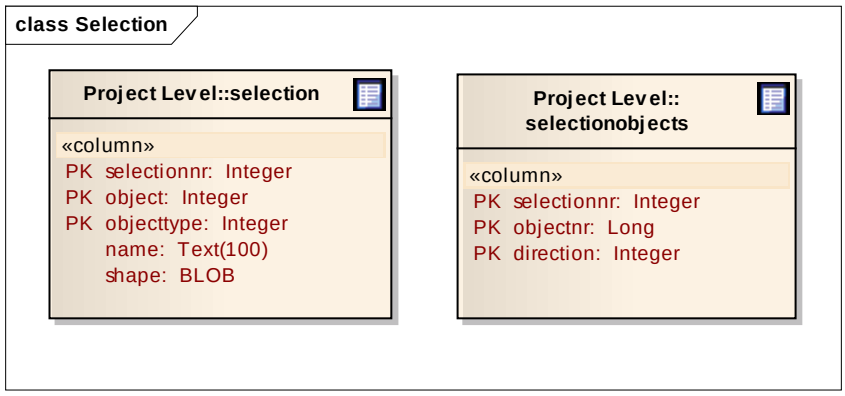

Figure 4: Selection Tables

Table Name: SELECTION

The table stores the available selection and the type of objects that the selection applies to. The type is determined by the object (point,line,triline,polyline or polygon) and the sub-type within that graphical object (objecttype). For a list of graphical objects and associated objecttypes, refer to the table in Section 2.4.

Field Name	Field type	Explanation
selectionnr	Integer	Number of the selection
Object	Short	Object shape 1: Point 2: Line 3: Triline 4: Polyline 5: Polygon
objecttype	Short	Type of the object per object shape: 1: centroid, link, turn, screenline, area 2: Node, transitline 4: Counts 5: Stops 6: Area-points
Name	String(100)	Name of the selection
shape	BLOB	Collection of points defining the geometry

Table Name: SELECTIONOBJECTS

The table stores the objects associated with a specific selection.

Field Name	Field type	Explanation
selectionnr	Integer	Number of the selection
objectnr	Integer	Number of the object
direction	Short	Direction of links (only used when the stored selection contains links)

3.1.5 Fares

The following tables are used to define the fares associated with public transport.

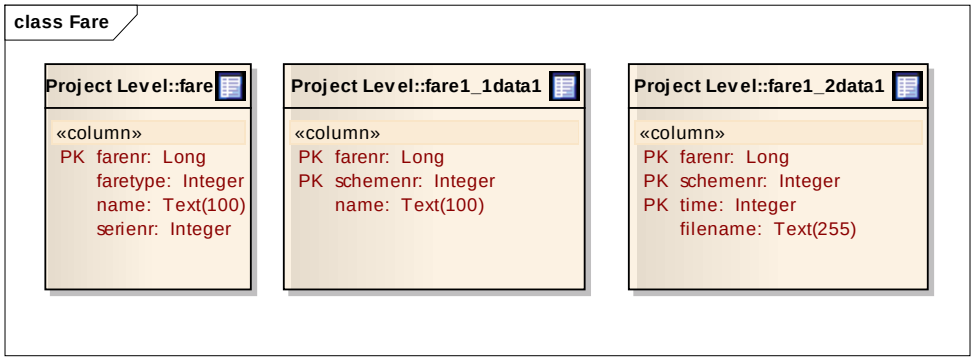

Table Name: FARE

This table stores the fares available in the project and their names.

Field Name	Field type	Explanation
farenr	Integer	Fare number
faretype	Short	Type of fare
name	String(100)	Name of the fare
serienr	Short	Reference to the type stored in the table SERIE

Table Name: FARE1_1DATA1

This table stores the various fare schemes and their names.

Field Name	Field type	Explanation
farenr	Integer	Fare number
schemenr	Short	Scheme Number
name	String(100)	Name of the scheme for a particular fare

Table Name: FARE1_2DATA1

This table stores the filenames where the interzonal fares are stored for particular scheme and time.

Field Name	Field type	Explanation
farenr	Integer	Fare number
schemenr	Short	Scheme Number
time	Short	Time dimension to which the fare and scheme applies
filename	String(255)	Name of the binary file storing the fares per zones

3.2 Level 1 and level 2 tables (the network)

The level 1 tables comprise the network. Per object there is a series of tables that make up the data for this object. Each object has a relation with a super network table at level 0 that define their position in the network.
For each object the most important table is the table with the name of the object (e.g. centroid or link). This table defines the presence of an object in the variant. For each object there can be one or more level 1 tables and one or more level 2 tables.

The tables for each object are shown in a diagram in which the object table is displayed in the middle. The relation with the super network is shown on the left side and the data tables are shown on the right side. Level 0 tables are shown in light pink, level 1 tables in cyan and level 2 tables in light green.

3.2.1 Centroid

Centroids are the origins and destinations of travel, representing the connections between the land use system (divided into zones) and the transport network. There is one centroid per zone.

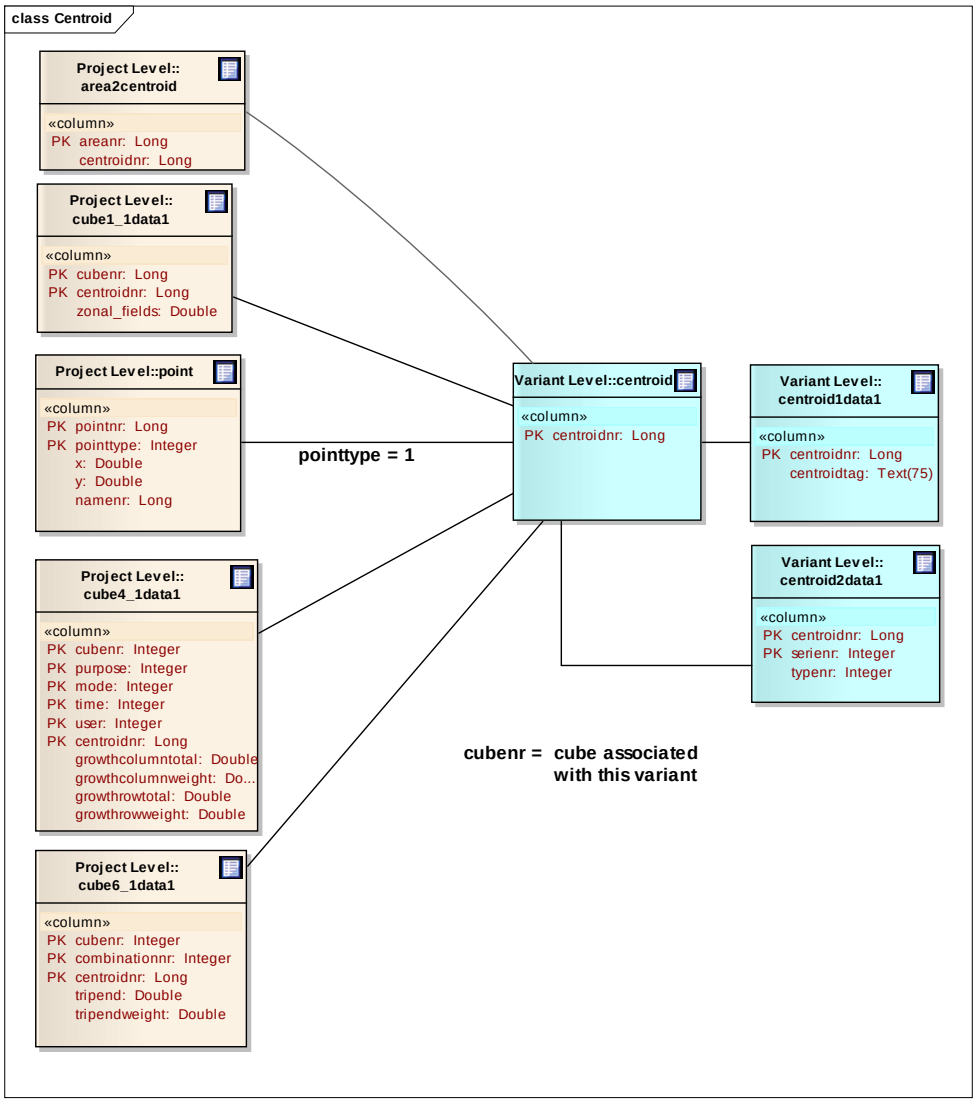

Figure 5: Centroid Tables

In the standard database, a centroid does not have level 2 data. A centroid does however have a special relation with two level 0 tables.

Table Name: CENTROID

This table contains the centroid number and the only purpose is to indicate if a centroid occurs in this variant.

Field Name	Field type	Explanation
centroidnr	Integer	Centroid number

Table Name: CENTROID1DATA1

This table contains centroid tags.

Field Name	Field type	Explanation
centroidnr	Integer	Centroid number
centroidtag	String(75)	Tag (text) for this centroid

This table is typically used to store any alternative centroid number, such as a hierarchical number.

Table Name: CENTROID2DATA1

This table contains the centroid types, i.e. the type-items for every defined type associated with centroids.

Field Name	Field type	Explanation
centroidnr	Integer	Centroid number
serienr	Short	Type number
typenr	Short	Type-item number

3.2.2 Node

Nodes are the points that connect the links.

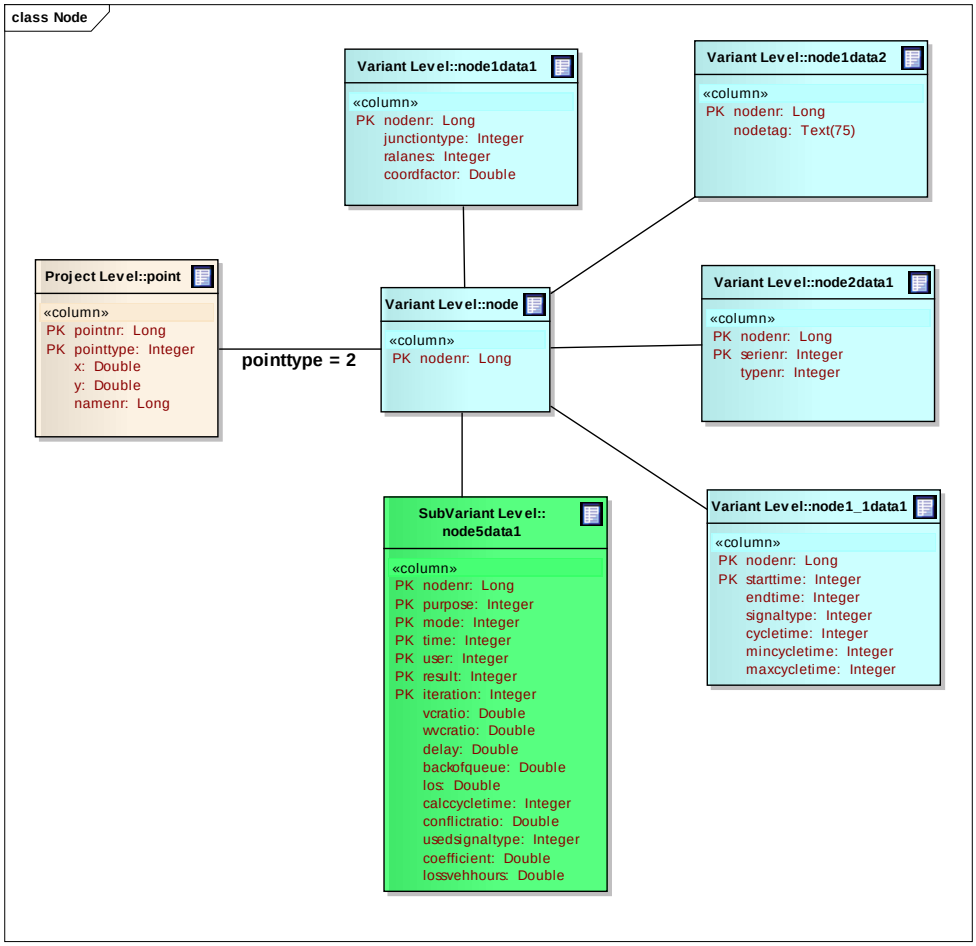

Figure 6: Node Tables

Table Name: NODE

This table contains the node number and the only purpose is to indicate if a node occurs in this variant.

Field Name	Field type	Explanation
nodenr	Integer	Node number

Table Name: NODE1DATA1

This table contains node junction data

Field Name	Field type	Explanation
nodenr	Integer	Node number
junctiontype	Short	Junction type: o: <undefined>

		1: Equal priority or uncontrolled junction 2: Give way (or two-way yield sign controlled, or priority) junction 3: Signalized junction 4: Roundabout 5: Signalized roundabout 6: All-way-stop controlled junction
ralanes	Short	Number of lanes on a roundabout for types 4, 5. Priority movement at the junction for types 2,6.
coordfactor	Float	Coordination or calibration factor for the junction. It is applied to the calculated delay for each turn at the modelled junction and is used mainly in combination with a static assignment model.

The presence of a record in this table for a node indicates that the node is a junction. A junction is defined with many tables. The entering links have data in the link1_1data1 table and the junction should also have turns defined to store the turning flows (turn and turn5data1).

Table Name: NODE1DATA2

This table contains an alternative number or tag for the node. This is typically used to keep the node numbering of some external network synchronized.

Field Name	Field type	Explanation
nodenr	Integer	Node number
nodetag	String(75)	Node tag

Table Name: NODE2DATA1

This table contains the type-items for every defined type associated with nodes.

Field Name	Field type	Explanation
nodenr	Integer	Node number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: NODE5DATA1

This table contains the node output data of the junction modelling.

Field Name	Field type	Explanation
nodenr	Integer	Node number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension

time	Short	Number of the time dimension
user	Short	Number of the user dimension
result	Short	Number of the result dimension
iteration	Short	Number of the iteration dimension
vcratio	Float	Critical volume capacity ratio for the whole junction
wvcratio	Float	Weighted volume capacity ratio for the whole junction
delay	Float	Weighted average total delay on a junction (delay due to the junction itself and the congestion).
backofqueue	Float	Weighted average queue length at the junction
los	Float	Level of service as set by the user. OmniTRANS does not itself calculate a level of service.
calccycletime	Short	Calculated cycle time. If the signal type is automated, the cycle time is actually calculated by the junction modelling module. If the signal type is manually coded (or actuated), this value is simply copied from the input to the output tables of OmniTRANS.
conflictratio	Float	Summation of the vcratio for all the movements composing the conflict group.
usedsignaltype	Short	Type of signal used during modelling at the junction. There are four signal types in OmniTRANS 1. Manual (fixed-time, pretimed) 2. Automated (optimised) 3. Actuated (cycle time actuated) 4. Actuated (vehicle-actuated)
lossvehhours	Float	Total vehicle hours lost at the node. This result is calculated by the dynamic class OtSeedPoint

Table Name: NODE1_1DATA1

This table stores information about the signal cycle time for a signalled junction.

Field Name	Field type	Explanation
nodenr	Integer	Node number
starttime	Short	Operational start time of the signal scheme
endtime	Short	Operational end time of the signal scheme
signaltype	Short	Signal types defined for the junction. There are four signal types in OmniTRANS 1. Manual (fixed-time, pretimed) 2. Automated (optimised)

		3. Actuated (cycle time actuated) 4. Actuated (vehicle-actuated)
cycletime	Short	The length of one signal cycle in seconds
mincycletime	Short	The minimum allowed cycle time (seconds), used for actuated signal control types. The minimum value must be a positive value.
maxcycletime	Short	The maximum allowed cycle time (seconds), used for actuated signal control types. The maximum value may not be lower than the average cycle time. The maximum value overrules the global maximum cycle time.

3.2.3 Count

Counts are used to represent sites that record count data on a link.

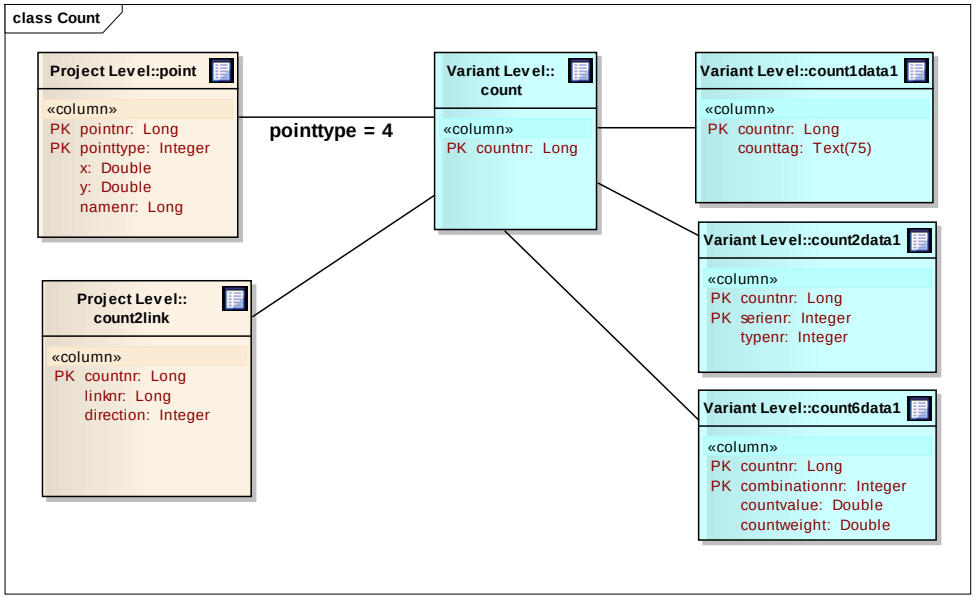

Figure 7: Count Tables

Table Name: COUNT

This table contains the count number and the only purpose is to indicate if a count occurs in this variant.

Field Name	Field type	Explanation
countnr	Integer	Count number

Table Name: COUNT1DATA1

This table contains an alternative number or tag for the count. This is typically used to keep the count numbering of some external network synchronized.

Field Name	Field type	Explanation
countnr	Integer	Count number
counttag	String(75)	Count tag

Table Name: COUNT2DATA1

This table contains the type-items for every defined type associated with counts.

Field Name	Field type	Explanation
countnr	Integer	Count number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: COUNT6DATA1

This table contains the count output data recorded by combination (production by purpose, attraction by purpose or screenline).

Field Name	Field type	Explanation
countnr	Integer	Count number
combinationnr	Short	Number of the combination
countvalue	Short	Value recorded for the count
countweight	Float	Weight associated with the count

3.2.4 Stop

Stops are points attached to nodes in the network that are used to define the route that transit lines take and locations where passengers can board, alight and transfer.

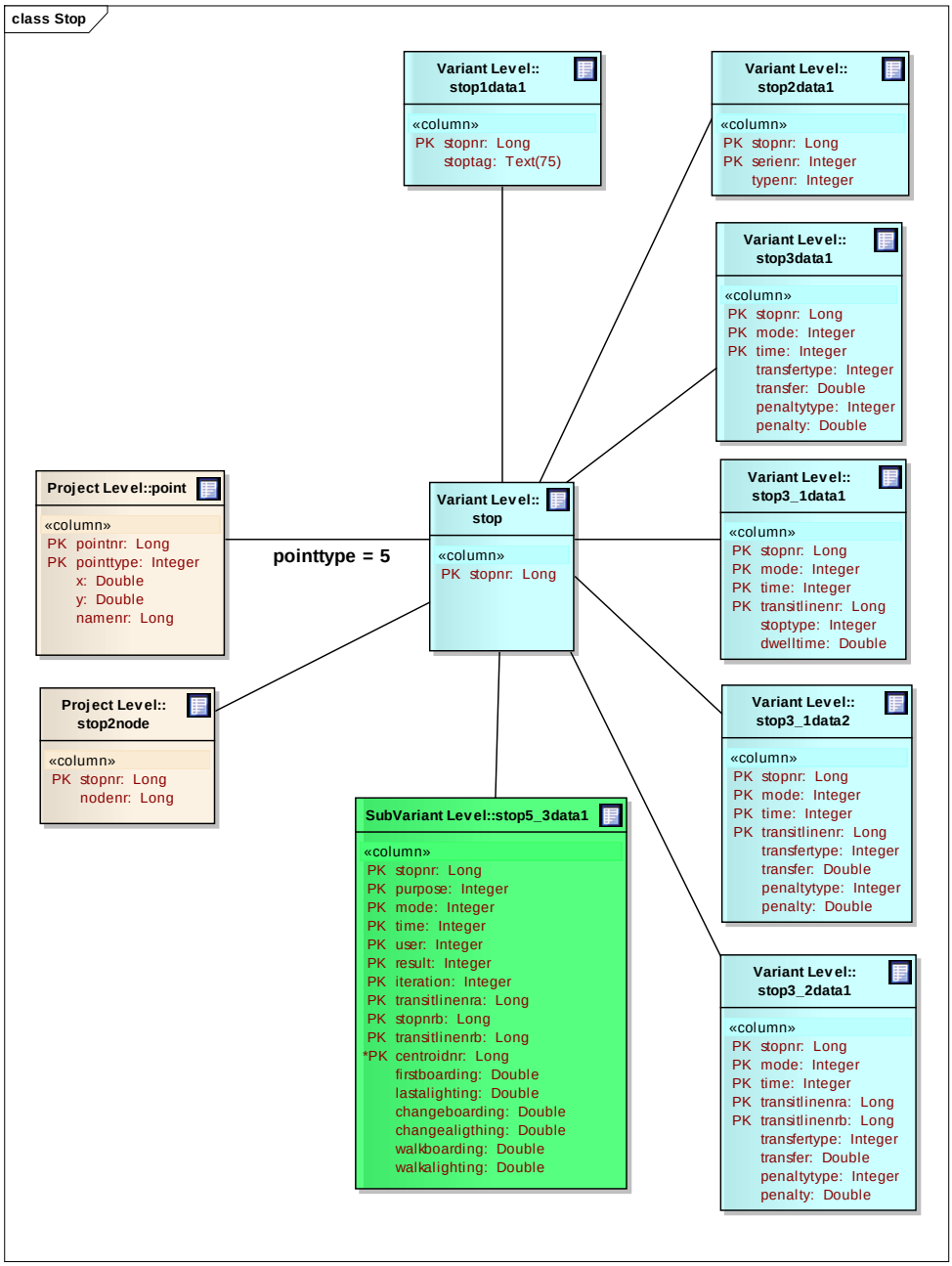

Figure 8: Stop Tables

Table Name: STOP

This table contains the stop number and the only purpose is to indicate if a stop occurs in this variant.

Field Name	Field type	Explanation
stopnr	Integer	Stop number

Table Name: STOP1DATA1

This table contains an alternative number or tag for the stop. This is typically used to keep the stop numbering of some external network synchronized.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
stoptag	String(75)	Stop tag

Table Name: STOP2DATA1

This table contains the type-items for every defined type associated with stops.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: STOP3DATA1

This table stores additional information on transfer and penalty associated with a stop. These can vary according to mode and time.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
transfertype	Short	Type of transfer applied to the transfer field: 1: Factor (multiply) 2: Constant (add)
transfer	Float	Transfer cost associated with this stop (factor or constant)
penaltytype	Short	Type of penalty applied to the penalty field: 1: Factor (multiply) 2: Constant (add)
penalty	Float	Penalty cost associated with this stop (factor or constant)

Table Name: STOP3_1DATA1

This table stores stop information dependent on a transit line, a mode and a time.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
transitlinenr	Integer	Number of the transit line
stoptype	Short	The type of stop. There are five possible types in OmniTRANS. 1. Normal 2. No transfer 3. No stop 4. Board only 5. Alight only
dwelltime	Float	Time that the transit vehicles remain at the stop before resuming their journey.

Table Name: STOP3_1DATA2

This table stores additional information on transfer and penalty associated with a stop for a specific transit line. These can vary according to mode and time.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
transitlinenr	Integer	Number of the transit line
transfertype	Short	Type of transfer applied to the transfer field: 1: Factor (multiply) 2: Constant (add)
transfer	Float	Transfer cost associated with this stop (factor or constant)
penaltytype	Short	Type of penalty applied to the penalty field: 1: Factor (multiply) 2: Constant (add)
penalty	Float	Penalty cost associated with this stop (factor or constant)

Table Name: STOP3_2DATA1

This table stores additional information on transfer and penalty associated with a transfer between two transit lines at a particular stop. These can vary according to mode and time.

Field Name	Field type	Explanation
stopnr	Integer	Stop number
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
transitlinenra	Integer	Number of the transit line a
transitlinenrb	Integer	Number of the transit line b
transfertype	Short	Type of transfer applied to the transfer field: 1: Factor (multiply) 2: Constant (add)
transfer	Float	Transfer cost associated with this stop and transit lines (factor or constant)
penaltytype	Short	Type of penalty applied to the penalty field: 1: Factor (multiply) 2: Constant (add)
penalty	Float	Penalty cost associated with this stop and transit lines (factor or constant)

Table Name: STOP5_4DATA1

This table contains the output data for the stop.

Field Name	Field type	Explanation
stopnr	Integer	Stop number (a)
PMTURI (6 fields)	Short	Number of the corresponding dimension
transitlinenra	Integer	Number of Transit line a
stopnrb	Integer	Number of stop for transit line b
transitlinenrb	Integer	Number of Transit line b
centroidnr	Integer	Identifier of the centroid object wrt to stopnr (a)
firstboarding	Float	Number of passengers in a zone for which the transit line represents their first boarding.
lastalighting	Float	Number of passengers per destination zone for which the transit line represents their last alighting.
changeboarding	Float	Number of passengers who board transitline a at stop a, having just transferred from transitline b at stop b. In this case, stop a and stop b are assumed to be co-located and that there is no substantial time required to walk between the two stops.
changealighting	Float	Number of passengers that are alighting from transitline a at stop a and that are destined for transitline b at stop b. In this case, stop a and stop

		b are assumed to be co-located and that there is no substantial time required to walk between the two stops.
walkboarding	Float	Number of passengers who are boarding transit line a at stop a, having just alighted transit line b at stop b, and walked from stop b to stop a. In this case, stop a and stop b are separated by a substantial distance and there is a walking time required to move from stop a to stop b.
walkalighting	Float	Number of passengers that are alighting from transit line a at stop a and are walking to stop b to board transit line b. In this case, stop a and stop b are separated by a substantial distance and there is a walking time required to move from stop a to stop b.

The key principle behind the field definitions in this table is that the "action" (i.e. walk boarding, walk alighting, change boarding, change alighting) is executed at stopnr(a), transitlinenra with stopnrb, transitlinenrb just being the "other related" stop and transit line.

When an interchange takes place (walking or not walking), there are always two records written to the database; one for the stop where people alight and one for the stop where people board.

Here is what happens when 20 people change at stop 123 from transit line 12 to 16. Line 12 has mode 30 and line 16 has mode 31. 20 people board line 16 and 20 people alight line 12. There are two records:

stopnr,p,m,t,u,r,i,transitlinenra,stopnrb,transitlinenrb,centroidnr,firstboarding,lastalighting, changeboarding, changealigting, walkbarding, walkalighting
123, p,31,t,u,r,i, 16, 0, 12, 0, 0, 0, 20, 0, 0, 0
123, p,30,t,u,r,i,12, 0, 16, 0, 0, 0, 0, 20, 0, 0

3.2.5 Area

Areas represent two dimensional surfaces and are mainly used to represent traffic zones. Areas can be associated with zone centroids.

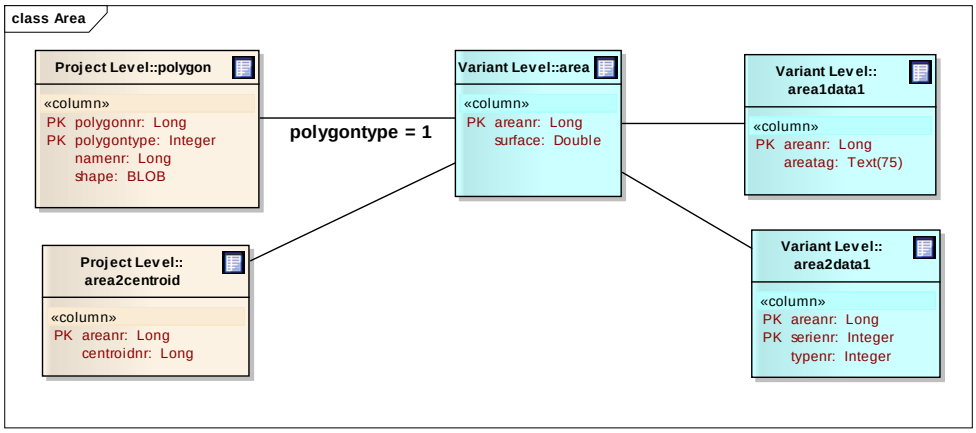

Figure 9: Area Tables

Table Name: AREA

This table contains the area number and the only purpose is to indicate if an area occurs in this variant.

Field Name	Field type	Explanation
areanr	Integer	Area number
surface	Float	Surface of the area as entered by the user in the unit of his choice. OmniTRANS provide a capability to calculate the area in square km.

Table Name: AREA1DATA1

This table contains an alternative number or tag for the area. This is typically used to keep the area numbering of some external network synchronized.

Field Name	Field type	Explanation
areanr	Integer	Area number
areatag	String(75)	Area tag

Table Name: AREA2DATA1

This table contains the type-items for every defined type associated with area.

Field Name	Field type	Explanation
countnr	Integer	Count number
serienr	Short	Type number
typenr	Short	Type-item number

3.2.6 Link

Links are the lines connecting nodes or centroids. A link is a transport channel connecting either two nodes, one centroid to a node or two centroids although not recommended modelling wise. A link can be divided into segments and the collection of segments/points composing the link are stored as one single entity, a BLOB, into the shape field. This allows the software to reduce the number of links in a transportation network while respecting the detailed geographical contour.

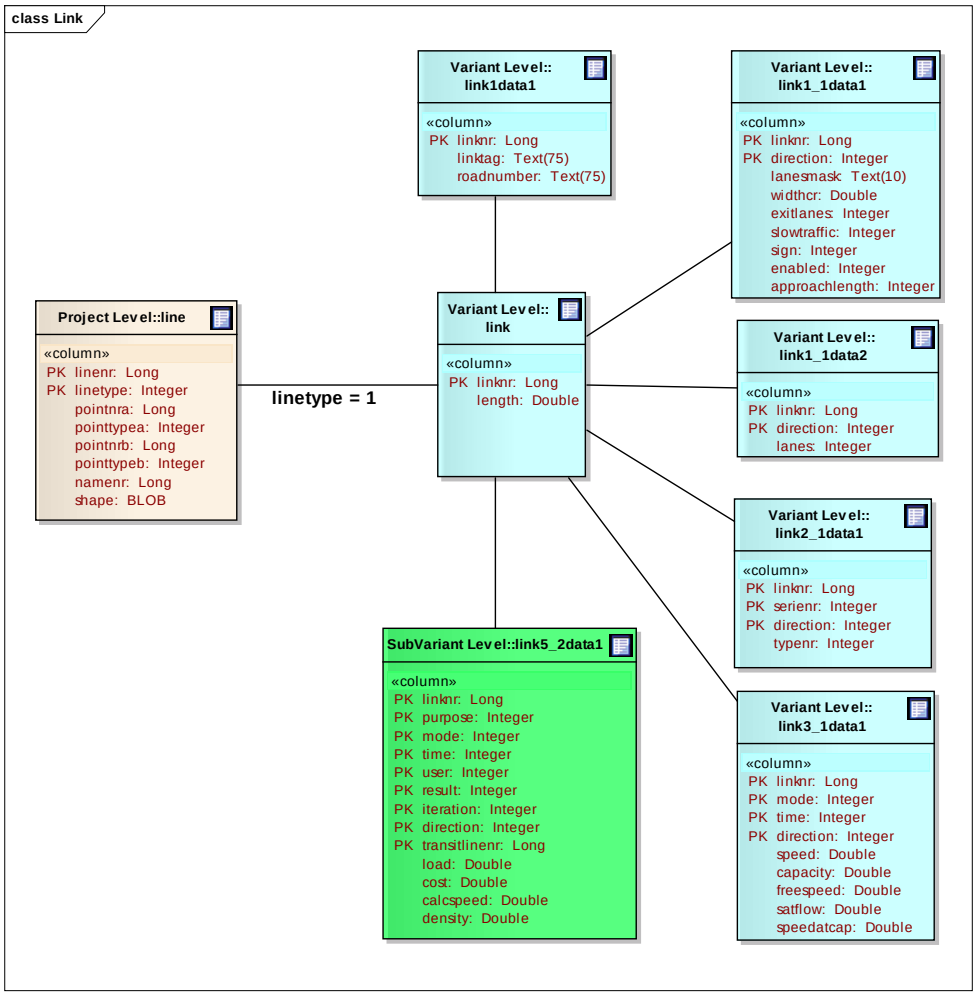

Figure 10: Link Tables

Table Name: LINK

This table contains the link number and the main purpose is to indicate if a link occurs in this variant.

Field Name	Field type	Explanation
linknr	Integer	Link number
length	Float	Length of the link in user defined units (km or

		miles)

Table Name: LINK1DATA1

This table contains an alternative number or tag for the link. This is typically used to keep the area numbering of some external network synchronized.

Field Name	Field type	Explanation
linknr	Integer	Link number
linktag	String(75)	Link tag
roadnumber	String(75)	Road number

Table Name: LINK1_1DATA1

This table contains link junction data

Field Name	Field type	Explanation
linknr	Integer	Link number
direction	Short	Direction of the link
lanesmask	String(10)	Approach lanes to a junction: A: All directions L: Left turn R: Right turn S: Straight P: Right + Straight Q: Left + Straight T: Left + Right
widthcr	Float	Width of the verge / shoulder (in meters)
exitlanes	Short	Number of lanes leaving the junction.
slowtraffic	Short	Presence of slow traffic (pedestrians) 0: No 1: Yes
sign	Short	Presence of a sign along the link: 0: Not 1: Yield sign 2: Stop sign
enabled	Short	Indicates if the link should be ignored for junction modelling. 0: Ignore 1: Enabled
approachlength	Short	Length of the approach lanes before reaching the destination junction (applicable for dynamic assignment).

The presence of a record in this table for a link indicates that the link approaches a junction. A junction is defined with many tables. The node has data on the type of the junction and the junction should also have turns defined to store the turning flows (turn and turn5data1).

Table Name: LINK1_1DATA2

This table stores the number of lanes for a link.

Field Name	Field type	Explanation
linknr	Integer	Link number
direction	Short	Direction of the link
lanes	Short	Number of lanes on the link

Table Name: LINK2_1DATA1

This table contains the type-items for every defined type associated with links. This table also contains the type-items for the special link type. The presence of a record for the link type for the given direction indicates for which modes the link is allowed. The absence of a record for the link type indicates that the link is closed for all traffic in that direction. So one-way links just don't have data for the opposite direction.

Field Name	Field type	Explanation
linknr	Integer	Link number
serienr	Short	Type number
direction	Short	Direction of the link
typenr	Short	Type-item number

Table Name: LINK3_1DATA1

This table stores the link parameters if set different than the ones specified by the default link type.

Field Name	Field type	Explanation
linknr	Integer	Link number
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
direction	Short	Direction of the link
speed	Float	Maximum speed on link
capacity	Float	Maximum capacity of link
freespeed	Float	Speed at free flow
satflow	Float	Saturation flow
speedatcap	Float	Speed at capacity

Table Name: LINK5_2DATA1

This table contains the link output data for the link. The table is also used to store transit line data per link.

Field Name	Field type	Explanation
linknr	Integer	Link number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
result	Short	Number of the result dimension
iteration	Short	Number of the iteration dimension
direction	Short	Direction of the link
transitlinenr	Integer	Transit line number (zero if none)
load	Float	Load for this link (volume) for the specified PMTURI and transit line
cost	Float	Generalised cost for the link, for the specified PMTURI and transit line. Note that the unit of measurement is dependent on the route factors set in the assignment algorithm.
calcspeed	Float	Average calculated speed
density	Float	Average density

3.2.7 Turn

Turns represent the directed movement through a node to which some form of data can be associated. Normally this is some form of impedance that represents the time penalty that is incurred in making that movement. It is defined by a from node (origin of movement), a through node (the current node through which the turn occurs) and a to node (destination of movement).

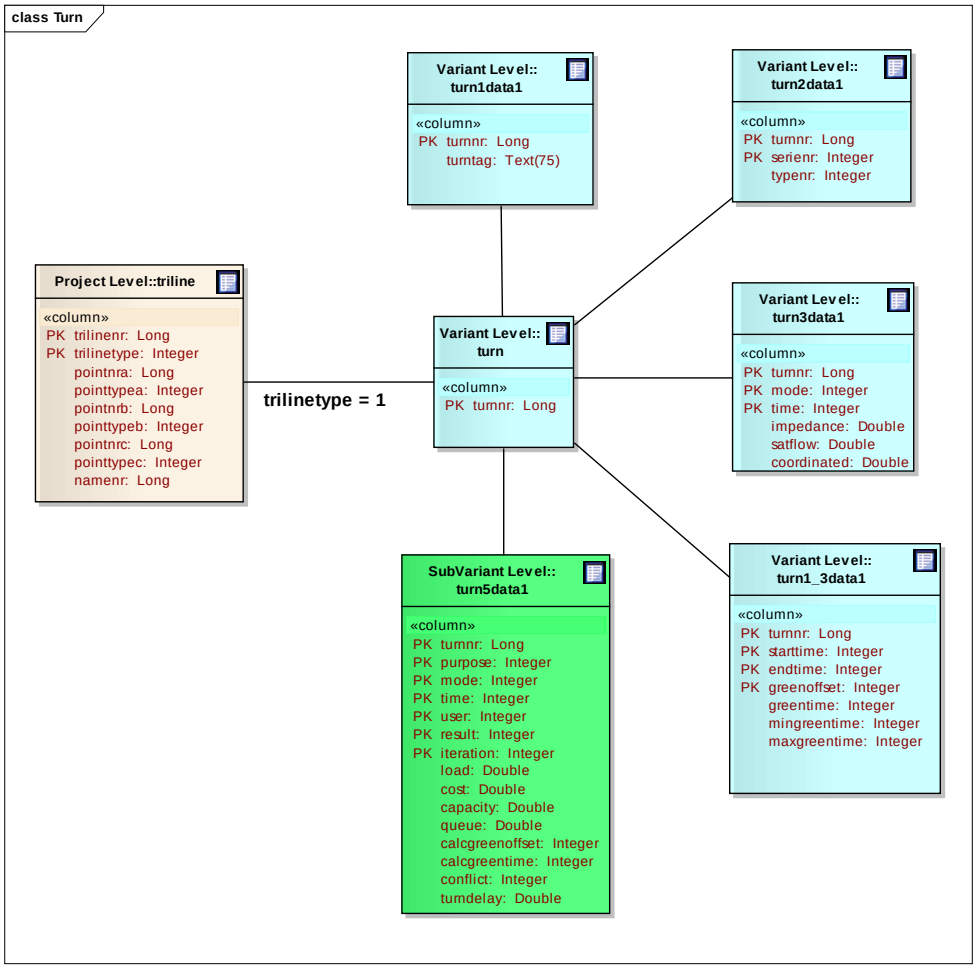

Figure 11: Turn Tables

Table Name: TURN

This table contains the turn number and the main purpose is to indicate if a turn occurs in this variant.

Field Name	Field type	Explanation
turnnr	Integer	Turn number

Table Name: TURN1DATA1

This table contains an alternative number or tag for the turn. This is typically used to keep the turn numbering of some external network synchronized.

Field Name	Field type	Explanation
turnnr	Integer	Turn number
turntag	String(75)	Turn tag

Table Name: TURN1_3DATA1

This table contains turn data dependent on the start, the end and the green offset times.

Field Name	Field type	Explanation
turnnr	Integer	Turn number
startime	Short	Operational start time for this signal scheme on the turn
endtime	Short	Operational end time for this signal scheme on the turn
greenoffset	Short	Offset from the start of the cycle time for the green time
greentime	Short	Time that the traffic light is green in a signal cycle.
mingreentime	Short	Minimum time for the green light
maxgreentime	Short	Maximum time for the green light

Table Name: TURN2DATA1

This table contains the turn types, i.e. the type-items for every defined type associated with turns.

Field Name	Field type	Explanation
turnnr	Integer	Turn number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: TURN3DATA1

This table stores the turn parameters dependent on network mode and time.

Field Name	Field type	Explanation
turnnr	Integer	Number of the turn
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
impedance	Float	Impedance due to the turn. If the impedance value is set to -1, the turn is banned for that particular mode and time. If the impedance value is set to zero or higher, it represents the average control delay on the turn for that particular mode and time. The unit of time is seconds. The impedance value overrules the delay value that is calculated by the junction modelling module.
satflow	Float	Saturation flow of turn. It is used to overrule the default saturation flow values that are used by the junction modelling module.
coordinated	Short	Indicates if the turn is coordinated with other turns. Default is not coordinated.

Table Name: TURN5DATA1

This table contains the turn output data.

Field Name	Field type	Explanation
turnnr	Integer	Turn number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
result	Short	Number of the result dimension
iteration	Short	Number of the iteration dimension
load	Float	Load for this turn (volume) for the specified PMTURI.
cost	Float	Generalised cost for this turn. The cost value represents the calculated average delay for a turning movement. The cost unit of measurement is dependent on the route factors set in the assignment algorithm.
capacity	Float	Maximum capacity for this turn. The capacity represents the maximum sustainable flow rate for a particular turning movement. The capacity includes effects of the type of movement, give

		way to other flows, blockade probabilities, number of lanes, sharing lanes, signal settings (ratio of cycle time and green time), etcetera. The units depend upon the chosen units for the OD-matrix that was assigned.
queue	Float	Queue represents the so called overflow queue. This is the average number of vehicles per cycle left over at the end of green periods at signals or at the end of acceptable gap (unblock) periods during gap-acceptance periods.
calcgreenoffset	Short	Green offset for this turn as calculated by the model. The calculation of this value is not yet implemented and is meant for future usage. The junction modelling module only determines the length of the green times, not the sequence en off sets for each signal group.
calcgreentime	Short	Calculated green time stored in seconds. If the signal type is automated, the green time is actually calculated by the junction modelling module. If the signal type is manually coded (or actuated), this value is simply copied from the input to the output tables of OmniTRANS.
conflict	Short	Indicate if this turn is part of the conflict group of movements at the junction.
turndelay	Float	Average turn delay in seconds as calculated by the junction modelling algorithm.

3.2.8 Screenline

Screenlines represent linked Count sites which are used in the Matrix Estimation process. They are also useful for generating reports where you want to compare observed vs assigned flows.

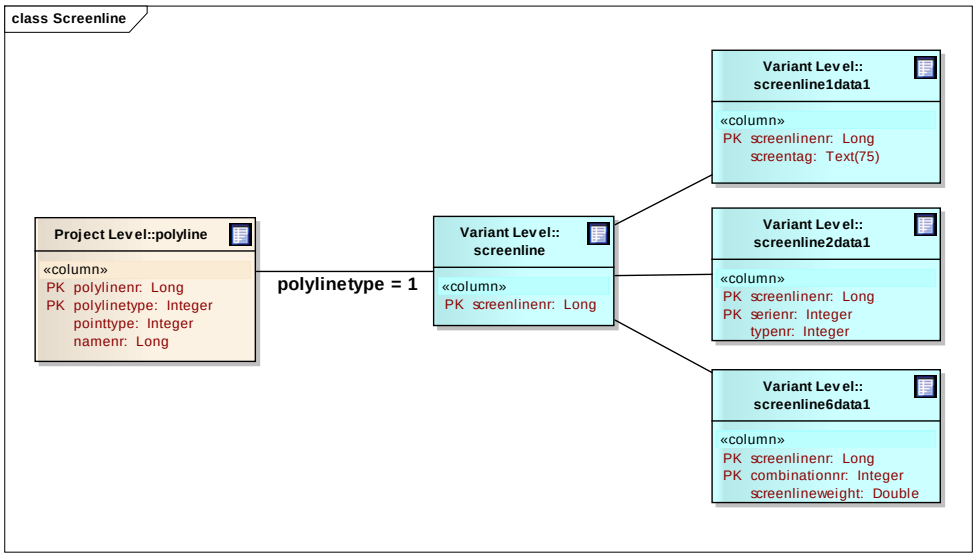

Figure 12: Screenline Tables

Table Name: SCREENLINE

This table contains the screenline number and the main purpose is to indicate if a screenline occurs in this variant.

Field Name	Field type	Explanation
screenlinenr	Integer	Screenline number

Table Name: SCREENLINE1DATA1

This table contains an alternative number or tag for the screenline. This is typically used to keep the screenline numbering of some external network synchronized.

Field Name	Field type	Explanation
screenlinenr	Integer	Screenline number
screenlinetag	String(75)	Screenline tag

Table Name: SCREENLINE2DATA1

This table contains the screenline types, i.e. the type-items for every defined type associated with screenlines.

Field Name	Field type	Explanation
screenlinenr	Integer	Turn number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: SCREENLINE6DATA1

This table stores the screenline weighting factor.

Field Name	Field type	Explanation
screenlinenr	Integer	Screenline number
combinationnr	Short	Number of the combination
screenlineweight	Float	Weight of the screenline

3.2.9 Transit Line

Transit lines define transit (public transport) services that traverse fixed routes through a network. The lines are defined in terms of their route, the speed/time at which the vehicles traverse the route, as well as their mode, capacity, headway etc.

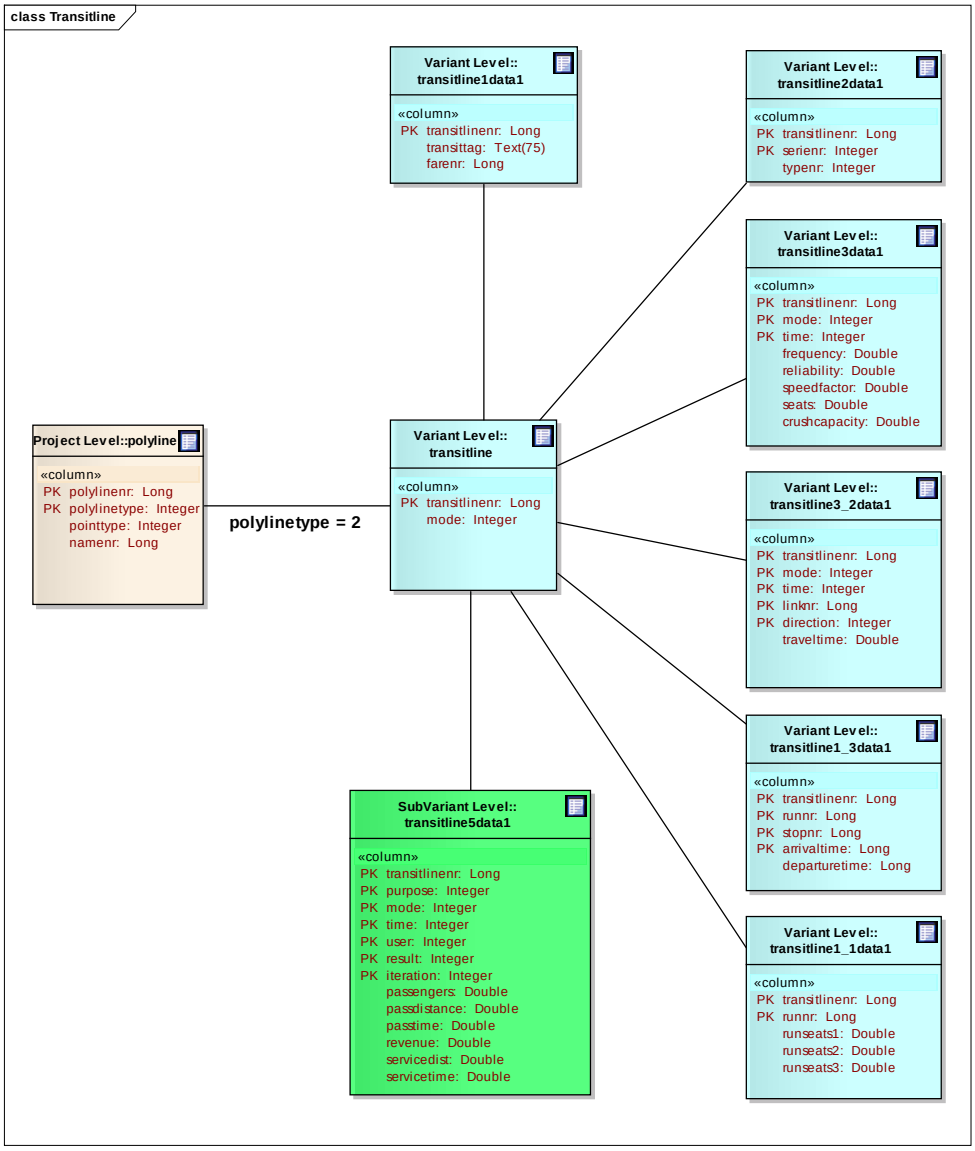

Figure 13: Transit Line Tables

Table Name: TRANSITLINE

This table contains the transit line number and its mode and the main purpose is to indicate if a transit line occurs in this variant.

Field Name	Field type	Explanation
transitlinenr	Short	Number of the transit line
mode	Short	Number of the mode dimension associated with the transit line

Table Name: TRANSITLINE1DATA1

This table contains an alternative number or tag for the transit line. This is typically used to keep the transit line numbering of some external network synchronized.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
transittag	String(75)	Transit line tag
farenr	Integer	Number of the fare associated with this transit line

Table Name: TRANSITLINE1_1DATA1

This table contains transit line data dependent on the run number.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
runnr	Short	Number of the run
status	Short	Indicates if a run is enabled (value = 1) or disabled (value=0)
runseats	Float	Number of seats available for the run.
runcrushcapacity	Float	Number of seats and standing places available for the run before the passengers experience crush in a public transport vehicle.

Table Name: TRANSITLINE1_3DATA1

This table contains transit line data dependent on the run number, the stop number and the arrival time.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
runnr	Integer	Number of the run
stopnr	Integer	Number of the stop
arrivaltime	Integer	Arrival time at the stop
departuretime	Integer	Departure time from the stop

Table Name: TRANSITLINE2DATA1

This table contains the transit line types, i.e. the type-items for every defined type associated with transit lines.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: TRANSITLINE3DATA1

This table stores the transit lines parameters that are dependent on network mode and time but independent of link and direction.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
frequency	Float	Frequency of the service (number of times per hour)
reliability	Float	Factor to indicate how reliable the frequency is. A factor 1.0 indicates 100% reliability.
speedfactor	Float	Factor to adjust the speed on the links.
seats	Float	Number of seats available.
crushcapacity	Float	Number of seats and standing places available before the passengers experience crush in a public transport vehicle. The crush capacity is used to define the crowding function of OtTransit.

Table Name: TRANSITLINE3_2DATA1

This table stores the transit lines parameters that are dependent on network mode and time as well as network link and direction.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
linknr	Integer	Number of the link
direction	Short	Direction of the link
traveltime	Float	Travel time for the transit line on the link

Table Name: TRANSITLINE5DATA1

This table contains the transit line output data.

Field Name	Field type	Explanation
transitlinenr	Integer	Number of the transit line
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
result	Short	Number of the result dimension
iteration	Short	Number of the iteration dimension
passengers	Float	Number of passengers using the transit line
passdistance	Float	Number of passengers*km using the transit line
passtime	Float	Total time that all passengers have spent on the transit line.
revenue	Float	Revenue generated by the transit line
servicedist	Float	Service distance which is equivalent to frequency *length.
servicetime	Float	Total operating service time which is equivalent to frequency * travel time where the travel time accounts for the speed factor, the congested speed factor, the dwell time, etc.

3.2.10 Control

Controls are point objects and can be attached to any other network objects. A control is an abstraction for objects or rules having an influence on or being influenced by other network objects. Examples include ramp metering installations, DRIPs, measuring equipment, lane restrictions, etc. The following design presents the minimum set of tables and fields available in the data dictionary definition of Version 6.0.

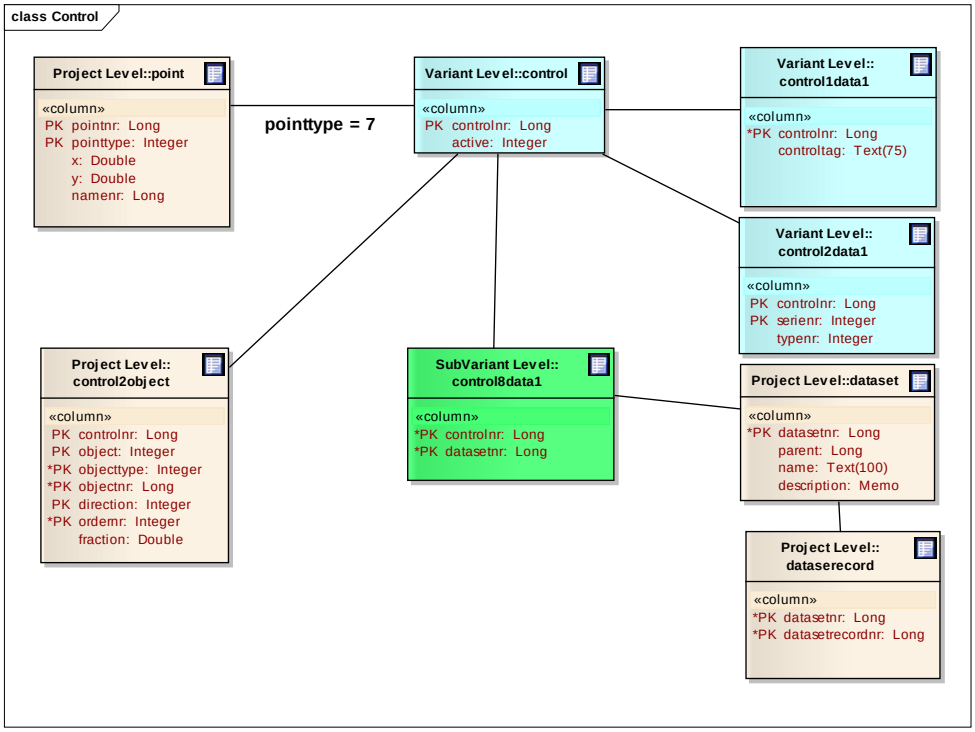

Figure 14: Control Tables

Table Name: CONTROL

This table contains the control number and the only purpose is to indicate if a control occurs in the variant.

Field Name	Field type	Explanation
controlnr	Integer	Control number
active	Short	Field indicating if the control is active or not. 0 if inactive, 1 if active

Table Name: CONTROL1DATA1

This table contains control tags.

Field Name	Field type	Explanation
controlnr	Integer	Control number
controltag	String(75)	Tag (text) for this control

This table is typically used to store any alternative control identifier. User-defined control attributes can be added in this table.

Table Name: CONTROL2DATA1

This table contains the type-items for every defined type associated with controls.

Field Name	Field type	Explanation
controlnr	Integer	Control number
serienr	Short	Type number
typenr	Short	Type-item number

Table Name: CONTROL8DATA1

This table saves the relations between controls and associated datasets .

Field Name	Field type	Explanation
controlnr	Integer	Route number
datasetnr	Integer	Dataset number

3.2.11 Route (User-defined)

A user-defined route represents a logical sequence of links and nodes. It is used to store a sequence of links connecting two different nodes in a network. The user-defined routes are graphical objects created by the user and they should not be confused with the paths/routes determined by the transport algorithms.

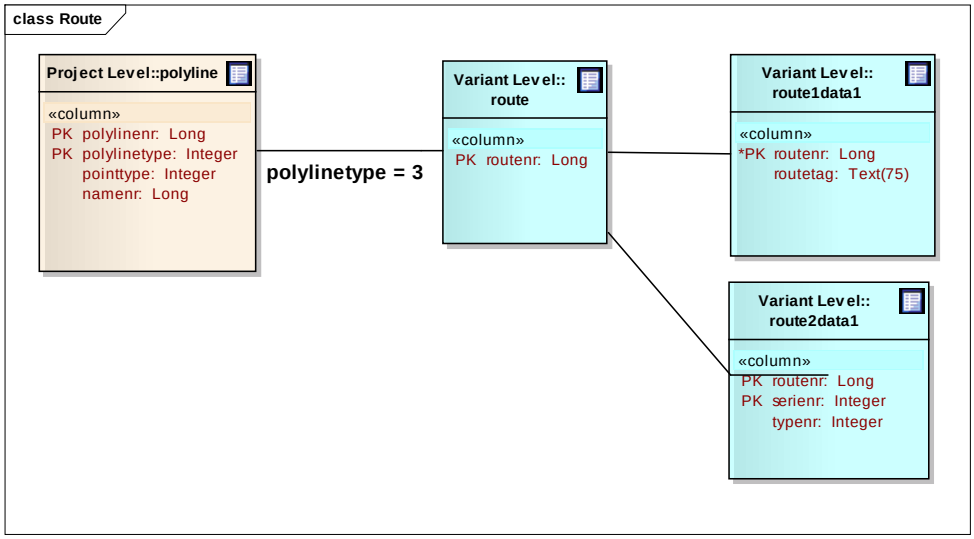

Figure 15: Route Tables

Table Name: ROUTE

This table contains the route number and the only purpose is to indicate if a route occurs in the variant.

Field Name	Field type	Explanation
routenr	Integer	Route number

Table Name: ROUTE1DATA1

This table contains route tags. This table is typically used to store any alternative route identifier. User-defined route attributes can be added in this table.

Field Name	Field type	Explanation
routenr	Integer	Route number
routetag	String(75)	Tag (text) for this route

Table Name: ROUTE2DATA1

This table contains the type-items for every defined type associated with routes.

Field Name	Field type	Explanation
routenr	Integer	Route number
serienr	Short	Type number
typenr	Short	Type-item number

3.2.12 Matrices

Matrices do not have a graphical representation. Per pmturi a filename is stored in the database (see Table CUBE and associates). The actual matrices are stored in binary files in the same directory as the variant.

Table Name: CUBE5DATA1

This table stores the filename containing the matrix data for a particular cube, purpose, mode, time, user, result and iteration.

Field Name	Field type	Explanation
cubenr	Integer	Matrix cube number
purpose	Short	Number of the purpose dimension
mode	Short	Number of the mode dimension
time	Short	Number of the time dimension
user	Short	Number of the user dimension
result	Short	Number of the result dimension
iteration	Short	Number of the iteration dimension
filename	String255	Filename of the binary file with the matrix data

The filename refers to a binary file that resides in the matrix directory. These binary matrices are stored in the same directory as the variant.

OD-matrices reside at level 0 tables, while other (generated) matrix types, such as skims and selected-links reside at level 2, because they are treated as output data.
The binary files have fixed extension. Per variant you can have a fixed number of output matrix cubes which have negative cube numbers:

Matrix type	cubenr	extension
Skim	-1	skm
Selected link	-2	slm
Cordon	-3	clm
Screenline	-4	scm
Transit	-5	stm

4 Audit Table

The application has an audit table that is not associated with any level (level is -1) and that is used to record the history of operations. The opening and closing of this table is handled differently than the other level 0, 1 and tables.

Table Name: Audit

Field Name	Field type	Explanation
timeinfo	String(50)	Time when the operation was performed
opstype	String(50)	Type of operation that was performed
variant	String(50)	Name of the main variant where the operation was performed
userinfo	String(50)	Name of the user who has performed the operation
machine	String(100)	Name of the machine where the operation was executed and the audit record registered
details	Memo	Stores the details of the operations associated with this audit record