ASTERA
DW Builder

ASTERA **DATA WAREHOUSE BUILDER**

A New Era in Data Warehousing

Astera
Enabling **Data-Driven** Innovation

Table of Contents

Overview

Data warehouses are backbone of enterprise data analytics. A high-quality data warehouse can provide tremendous competitive advantage to an enterprise by helping them derive meaningful insights from data spread across the enterprise.

Building and managing data warehouses is hard. Many data warehouse projects fail to complete. Most others underdeliver, are delayed, or go over budget. Despite these drawbacks, the enormous value delivered by successful data warehousing projects compels organizations to invest in these projects.

Astera Data Warehouse Builder (ADWB) is a no-code platform that accelerates development and modernization of data warehouses by featuring a metadata driven approach to building and managing data warehouses. ADWB can accelerate your data warehouse projects by as much as tenfold by eliminating and automating bulk of the tasks associated with building and managing data warehouses. ADWB supports dimensional and data vault 2.0 data warehouses.

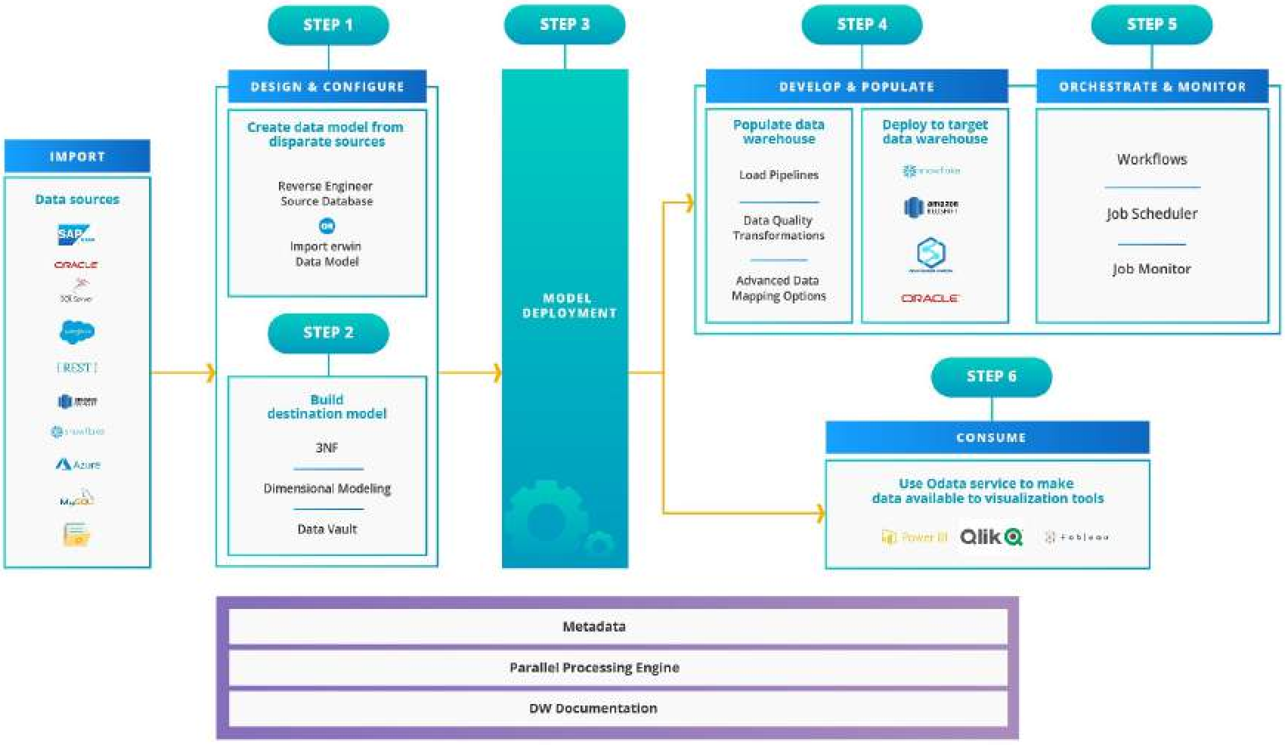
This document provides an overview of ADWB, discusses the key concepts and technologies, and provides insights into how the product reduces the complexity of building a data warehouse.

A new approach is needed

Astera brings a whole new approach to building a data warehouse. Based on an enriched data model, this approach is designed to automate and, where possible, eliminate many of the time-intensive, development tasks required to build and manage a data warehouse.

ADWB offers an iterative, agile model for end-to-end data warehousing. This compresses the feedback loop and enables incremental reviews and delivery of the functionality. Short feedback loops speed up projects, ensure that users' feedback is incorporated in a timely fashion and ensure that the product aligns with users' needs.

At the same time, the no-code environment substantially lowers the technical skill bar required for building and operating a data warehouse. This has a major impact on whether and how an organization builds a data warehouse. With this environment, data experts and analysts can build an entire data warehouse architecture without relying on a diverse set of technical skills and tools—otherwise needed for data warehousing projects.



Model driven warehouse development

Data model driven warehouse development—pioneered by Astera—puts data model at the front and center of a data warehouse project. This approach uses specialized data models to build and manage data stores. These specialized models are enriched to contain properties that are used to load and manage various aspects of a warehouse. Examples of these specialized models include dimensional model, data vault 2.0, and standard OLTP models.

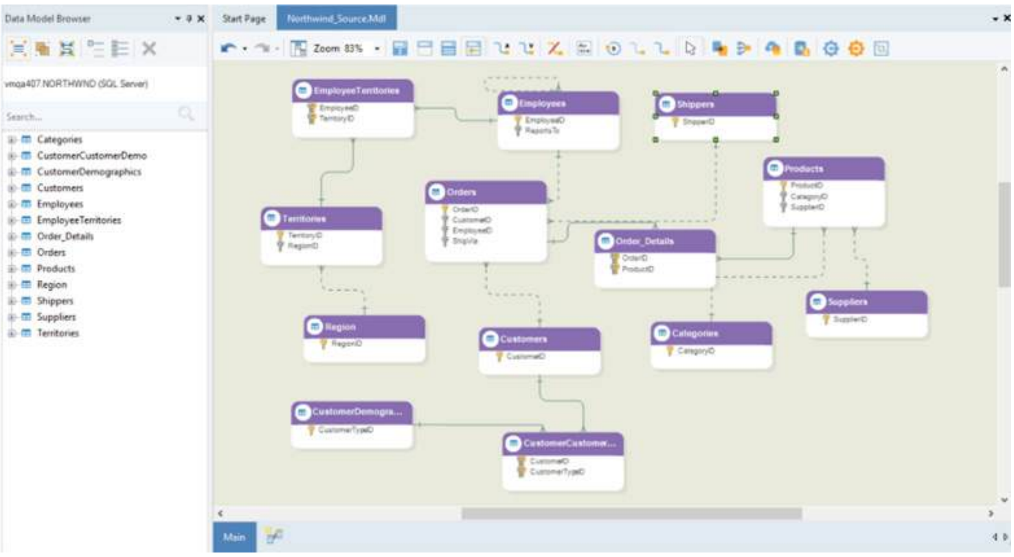
A dimensional model, for instance, contains detailed metadata about dimensions, facts, aggregates, and cubes. This metadata forms the foundation of Astera's no-code approach. At load or query times, these properties control dimension data loading and dimension lookup during fact loading. Similar checks ensure correctness of OLTP, data vault, and other model types.

ADWB has hundreds of model integrity checks to ensure that data model is correctly designed. This helps ensure the consistency of data model and helps identify and correct any errors early enough to avoid any inconsistencies and inaccuracies during warehouse population and maintenance.

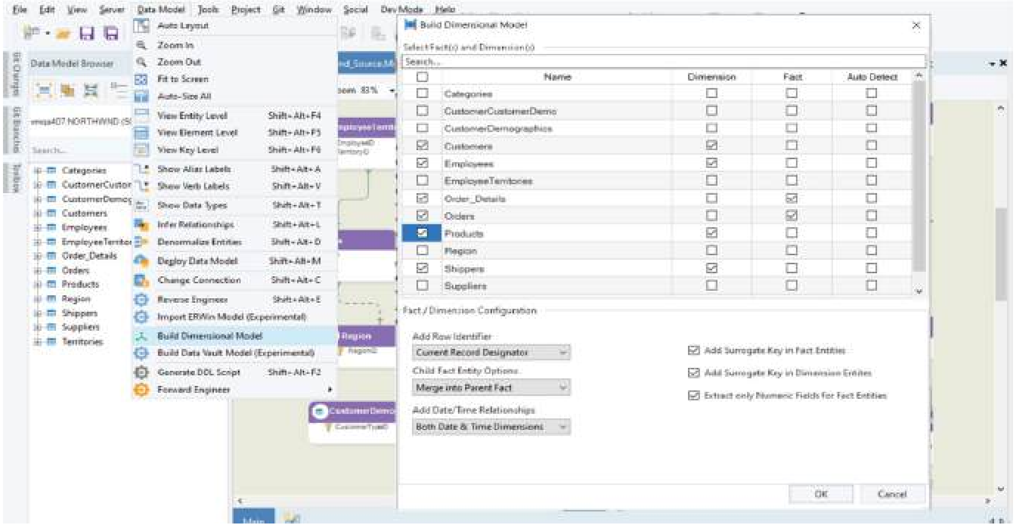
Designing a warehouse data model

Data warehouse builder typically starts by building a data model for the warehouse. This model can be built in Astera's data model designer, imported from another tool like ERWin, or reverse engineered from an existing database if working with an existing data warehouse.

ADWB features a data model designer where users can design special purpose data models. These include dimensional, data vault, and OLTP models. The designer enables creation of specific dimensional entities such as fact, dimension, and aggregates. Additionally, users can define specific roles for attributes that control loading of slowly changing dimensions and dimension lookups.



The designer provides the ability to derive a dimensional model from an OLTP model. These features enable rapid creation of a dimensional model from a source data model.


File Edit View Server Data Model Tools Project Git Window Social Dev Mode Help
Auto Layout
Zoom In
Zoom Out
Fit to Screen
Auto-Size All
View Entity Level Shift+Alt+F4
View Element Level Shift+Alt+F5
View Key Level Shift+Alt+F6
Show Alias Labels Shift+Alt+A
Show Verb Labels Shift+Alt+V
Show Data Types Shift+Alt+T
Infer Relationships Shift+Alt+L
Denormalize Entities Shift+Alt+D
Deploy Data Model Shift+Alt+M
Change Connection Shift+Alt+C
Reverse Engineer Shift+Alt+E
Import ERWin Model (Experimental)
Build Dimensional Model
Build Data Vault Model (Experimental)
Generate DDL Script Shift+Alt+F2
Forward Engineer
Data Model Browser
vmqa407.NORTHWND (S
Search...
Main

Build Dimensional Model

Select Fact(s) and Dimension(s)

Search...

□	Name	Dimension	Fact	Auto Detect
□	Categories	□	□	□
□	CustomerCustomerDemo	□	□	□
□	CustomerDemographics	□	□	□
☑	Customers	☑	□	□
☑	Employees	☑	□	□
□	EmployeeTerritories	□	□	□
☑	Order_Details	□	☑	□
☑	Orders	□	☑	□
☑	Products	☑	□	□
□	Region	□	□	□
☑	Shippers	☑	□	□
□	Suppliers	□	□	□

Fact / Dimension Configuration

Add Row Identifier: Current Record Designator

Child Fact Entity Options: Merge into Parent Fact

Add Date/Time Relationships: Both Date & Time Dimensions

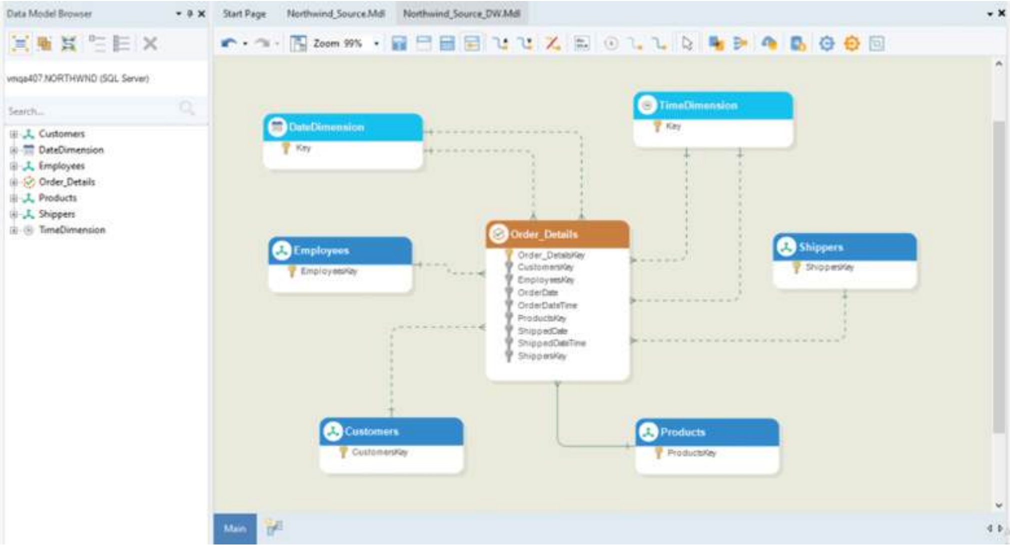
☑ Add Surrogate Key in Fact Entities

☑ Add Surrogate Key in Dimension Entities

☑ Extract only Numeric Fields for Fact Entities

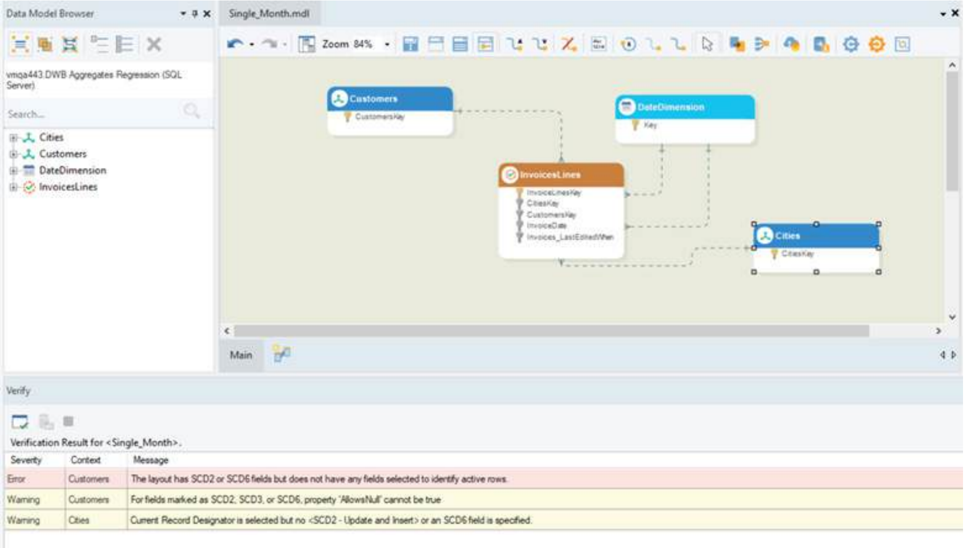
OK Cancel

Bulk of the work required to build a data warehouse is done in the data model designer. Once the model is designed and verified, user can create the a physical schema in an existing database using the forward engineering capability. At this point, the data warehouse is ready for loading.



Data Model Verification

Data warehouses can get quite large and complex as they evolve. As schema evolves, it becomes harder to maintain consistency across multiple entities and within a single entity. Some dimension entities can become quite complex once multiple types of slowly changing dimension fields are introduced. To ensure that a model is semantically correct and consistent, ADWB provides verification features that perform hundreds of checks. These checks include extensive validations of entities, relationships, aggregates, and more.



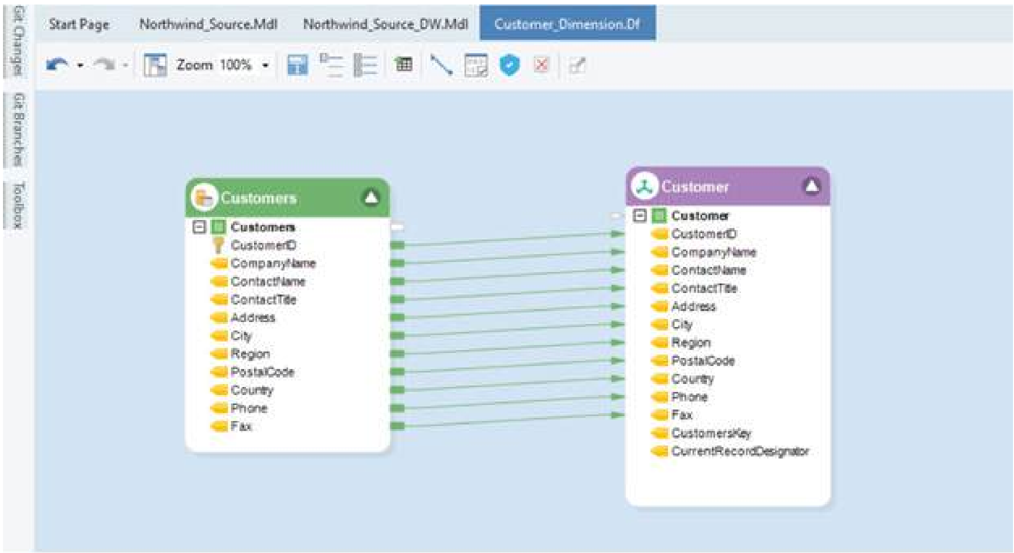
Verification Result for <Single_Month>.

Severity	Context	Message
Error	Customers	The layout has SCD2 or SCD6 fields but does not have any fields selected to identify active rows.
Warning	Customers	For fields marked as SCD2, SCD3, or SCD6, property 'AllowsNull' cannot be true
Warning	Cities	Current Record Designator is selected but no <SCD2 - Update and Insert> or an SCD6 field is specified.

Loading a dimensional data warehouse

Loading a data warehouse with hand-coded SQL statements is complicated and cumbersome. Because of their complexity, these SQL statements require experienced database developers to write and debug. Such resources are difficult to recruit, train, and retain. As the warehouse evolves, these queries become more and more expensive to maintain. The upshot is that changes become increasingly costly and time consuming.

This is where the true power of Astera's approach manifests itself. The dimensional model created by the user contains most of the information needed to load a data warehouse. Most of the complexities of loading dimensions and facts are reduced to simple source to target maps.



ADWB is built on Astera's award winning Centerprise platform that supports connectivity to a wide range of data sources including file formats, databases, ERP/CRM apps, and services. The platform supports cloud and on-premises databases and file systems.

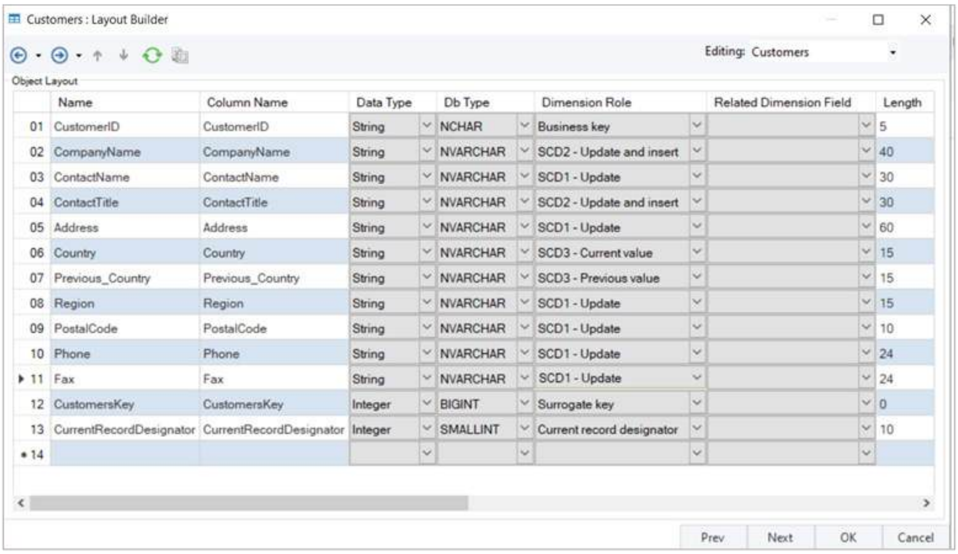
Managing dimensions

Managing dimensions—especially the ones that support history—is a complex undertaking. In a data warehouse, loading data into dimensions usually require multiple complex SQL commands. With ADWB, dimension loading is controlled via metadata in the dimensional model.

When designing data models, users specify roles for entities—facts, dimensions, aggregates. Additionally, attributes are also assigned roles. For instance, in dimension entities, attributes can be designated as Surrogate Keys, Business Keys, Effective and expiration dates, and more.

Slowly changing dimensions

ADWB supports most slowly changing dimension types like 1, 2, 3, and 6. Slowly changing dimensions enable users to maintain history of changes to dimensions over time. For more information on slowly changing dimensions, see here. These roles are used by the loading engine to automatically manage dimensions tables.



Customers : Layout Builder

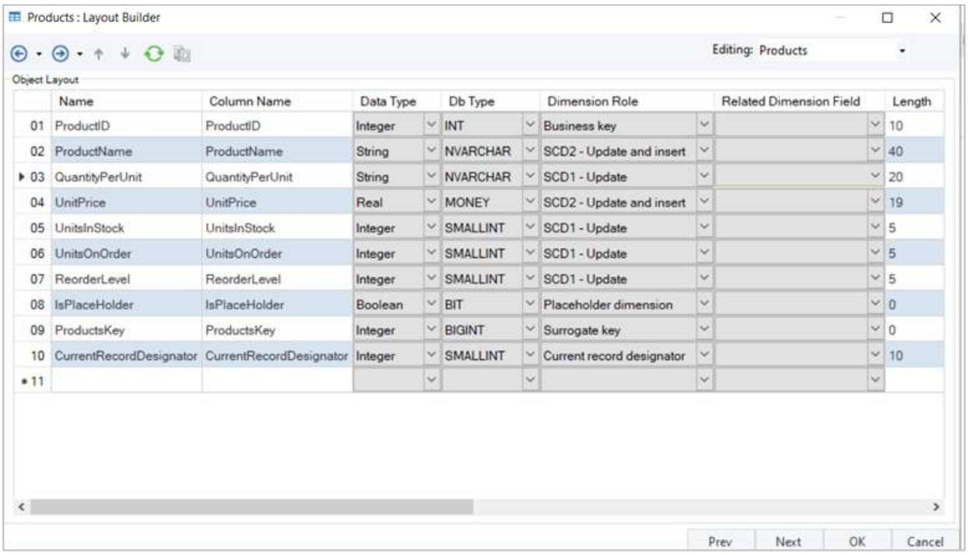
Editing: Customers

Object Layout

	Name	Column Name	Data Type	Db Type	Dimension Role	Related Dimension Field	Length
01	CustomerID	CustomerID	String	NCHAR	Business key		5
02	CompanyName	CompanyName	String	NVARCHAR	SCD2 - Update and insert		40
03	ContactName	ContactName	String	NVARCHAR	SCD1 - Update		30
04	ContactTitle	ContactTitle	String	NVARCHAR	SCD2 - Update and insert		30
05	Address	Address	String	NVARCHAR	SCD1 - Update		60
06	Country	Country	String	NVARCHAR	SCD3 - Current value		15
07	Previous_Country	Previous_Country	String	NVARCHAR	SCD3 - Previous value		15
08	Region	Region	String	NVARCHAR	SCD1 - Update		15
09	PostalCode	PostalCode	String	NVARCHAR	SCD1 - Update		10
10	Phone	Phone	String	NVARCHAR	SCD1 - Update		24
▸ 11	Fax	Fax	String	NVARCHAR	SCD1 - Update		24
12	CustomersKey	CustomersKey	Integer	BIGINT	Surrogate key		0
13	CurrentRecordDesignator	CurrentRecordDesignator	Integer	SMALLINT	Current record designator		10
* 14							

Late arriving dimensions

Dimension loader also supports placeholder dimension records for late arriving dimensions. This is necessary when facts arrive for a business key before a dimension record has been created. When the dimension arrives later, it is updated and the link to fact records is preserved.



Products : Layout Builder

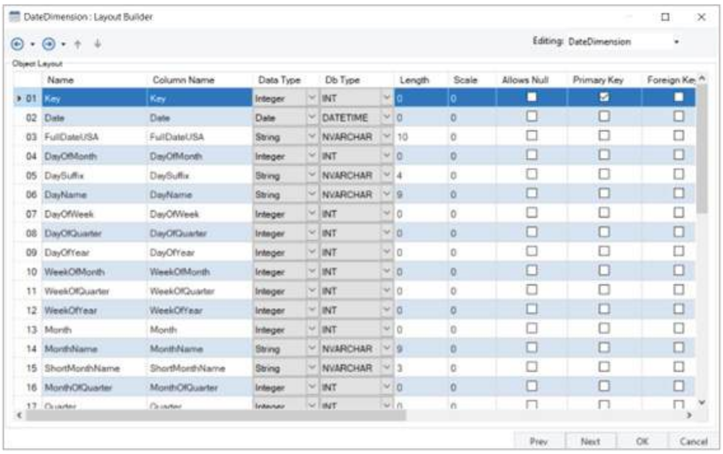
Editing: Products

Object Layout

	Name	Column Name	Data Type	Db Type	Dimension Role	Related Dimension Field	Length
01	ProductID	ProductID	Integer	INT	Business key		10
02	ProductName	ProductName	String	NVARCHAR	SCD2 - Update and insert		40
▸ 03	QuantityPerUnit	QuantityPerUnit	String	NVARCHAR	SCD1 - Update		20
04	UnitPrice	UnitPrice	Real	MONEY	SCD2 - Update and insert		19
05	UnitsInStock	UnitsInStock	Integer	SMALLINT	SCD1 - Update		5
06	UnitsOnOrder	UnitsOnOrder	Integer	SMALLINT	SCD1 - Update		5
07	ReorderLevel	ReorderLevel	Integer	SMALLINT	SCD1 - Update		5
08	IsPlaceHolder	IsPlaceHolder	Boolean	BIT	Placeholder dimension		0
09	ProductsKey	ProductsKey	Integer	BIGINT	Surrogate key		0
10	CurrentRecordDesignator	CurrentRecordDesignator	Integer	SMALLINT	Current record designator		10
* 11							

Date and time dimensions

ADWB has built-in date and time dimensions to support temporal data. These dimensions contain attributes to analyze and visualize data based on a large contingent of time periods including by day, weekday, day of month, day or year, and much more. Time dimension can help you plot and analyze data by time of day. So if, a retail store wants to analyze foot traffic based on time of day, time dimension can be used for it.

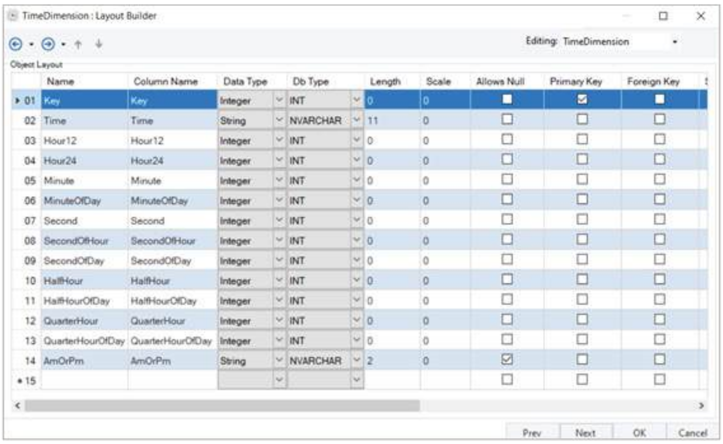


DateDimension : Layout Builder

Editing: DateDimension

Object Layout

	Name	Column Name	Data Type	Db Type	Length	Scale	Allows Null	Primary Key	Foreign Ke
▸ 01	Key	Key	Integer	INT	0	0	□	☑	□
02	Date	Date	Date	DATETIME	0	0	□	□	□
03	FullDateUSA	FullDateUSA	String	NVARCHAR	10	0	□	□	□
04	DayOfMonth	DayOfMonth	Integer	INT	0	0	□	□	□
05	DaySuffix	DaySuffix	String	NVARCHAR	4	0	□	□	□
06	DayName	DayName	String	NVARCHAR	9	0	□	□	□
07	DayOfWeek	DayOfWeek	Integer	INT	0	0	□	□	□
08	DayOfQuarter	DayOfQuarter	Integer	INT	0	0	□	□	□
09	DayOfYear	DayOfYear	Integer	INT	0	0	□	□	□
10	WeekOfMonth	WeekOfMonth	Integer	INT	0	0	□	□	□
11	WeekOfQuarter	WeekOfQuarter	Integer	INT	0	0	□	□	□
12	WeekOfYear	WeekOfYear	Integer	INT	0	0	□	□	□
13	Month	Month	Integer	INT	0	0	□	□	□
14	MonthName	MonthName	String	NVARCHAR	9	0	□	□	□
15	ShortMonthName	ShortMonthName	String	NVARCHAR	3	0	□	□	□
16	MonthOfQuarter	MonthOfQuarter	Integer	INT	0	0	□	□	□
17	Quarter	Quarter	Integer	INT	0	0	□	□	□



TimeDimension : Layout Builder

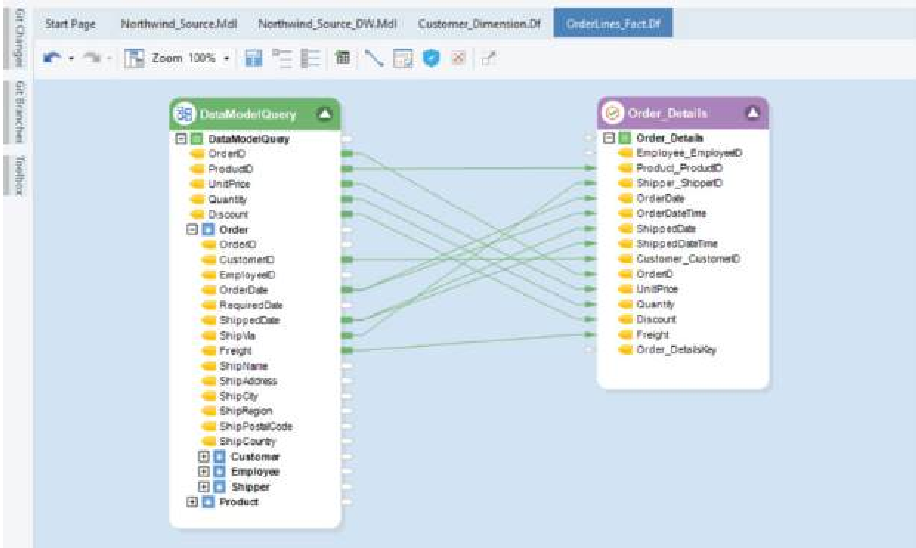
Editing: TimeDimension

Object Layout

	Name	Column Name	Data Type	Db Type	Length	Scale	Allows Null	Primary Key	Foreign Key
▸ 01	Key	Key	Integer	INT	0	0	□	☑	□
02	Time	Time	String	NVARCHAR	11	0	□	□	□
03	Hour12	Hour12	Integer	INT	0	0	□	□	□
04	Hour24	Hour24	Integer	INT	0	0	□	□	□
05	Minute	Minute	Integer	INT	0	0	□	□	□
06	MinuteOfDay	MinuteOfDay	Integer	INT	0	0	□	□	□
07	Second	Second	Integer	INT	0	0	□	□	□
08	SecondOfHour	SecondOfHour	Integer	INT	0	0	□	□	□
09	SecondOfDay	SecondOfDay	Integer	INT	0	0	□	□	□
10	HalfHour	HalfHour	Integer	INT	0	0	□	□	□
11	HalfHourOfDay	HalfHourOfDay	Integer	INT	0	0	□	□	□
12	QuarterHour	QuarterHour	Integer	INT	0	0	□	□	□
13	QuarterHourOfDay	QuarterHourOfDay	Integer	INT	0	0	□	□	□
14	AmOrPm	AmOrPm	String	NVARCHAR	2	0	☑	□	□
• 15							□	□	□

Loading facts

Loading data into fact tables involves looking up the appropriate dimension row for a business key. Dimensions with history—those having slowly changing dimension types 2, 3, or 6—have multiple rows for the same business key and the loader must select the correct row based on user defined criteria such as effective/expiry date, version number, or current record flag. The loader must also correctly process records for business keys where a dimension record is not yet created.

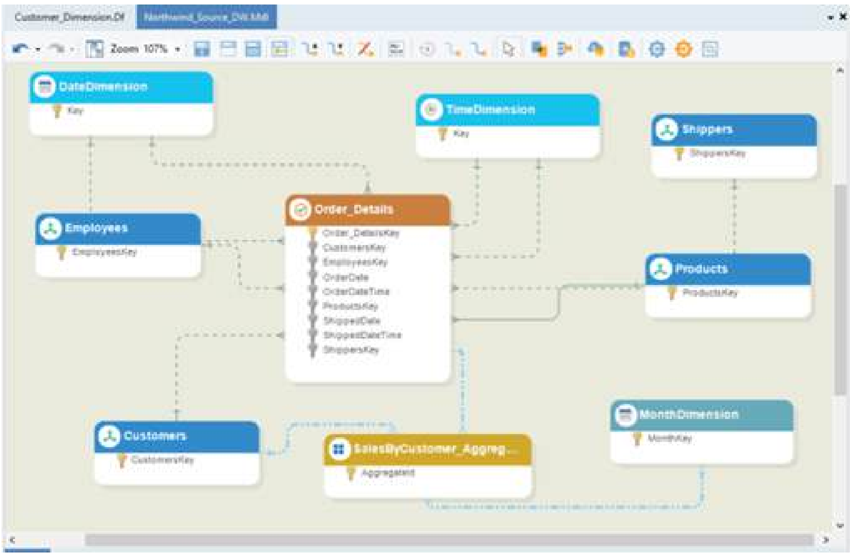


Astera
Enabling **Data-Driven** Innovation

Here again, data model driven approach pays off. Using a combination of data model relationships and dimension metadata, ADWB builds and executes the commands for the underlying databases. If a dimension record for a business key has not arrived yet, ADWB creates a placeholder dimension record. These placeholder dimensions are replaced with actual record during a subsequent dimension load. Moreover, if the fact table has missing entries that do not require the active surrogate key for a particular dimension record, ADWB uses the transaction date key to lookup the record that was active during a particular effective-expiration date range.

Managing aggregates

With ADWB, you can define aggregate tables in the data model. Aggregate tables can accelerate query performance by precomputing data for selected dimensions. Aggregate tables are summaries of granular fact tables based on specific dimension keys. For instance, we may compute aggregate of a sales fact table by creating a sum of all the sales for a product/branch combination by day, hour of day, or any other combination of dimensions. With ADWB, you can automatically update aggregate whenever new records are loaded in the underlying fact table.



Pushdown Optimization

Timely availability of data is a key component of a data warehouse. Therefore, fast loading of data is paramount for an enterprise data warehouse.

AWDB designers have paid a great deal of attention to loading of data into a data warehouse. This includes, but is not limited to, fast loading of data from data files, pushdown optimizations to perform much of dimension management, dimension lookups for fact loading, and fact aggregation is performed using commands that are optimized for the underlying database. More information about Astera's pushdown optimization technology is found here.

Data quality and verification

ADWB's load functionality is designed to ensure that data that is loaded in the warehouse is consistent and clean. If, however, you are modernizing an existing data warehouse and want to ensure that data in the warehouse is consistent and confirms to the warehouse model, you can use Verification features of ADWB. This feature performs many kinds of checks including referential integrity and domain validations.

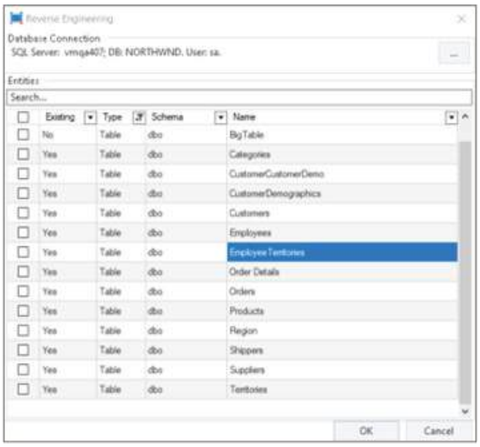
Connecting with sources

ADWB is built on top of Astera's enterprise-caliber ETL and ELT engines. These engines are capable of efficiently processing large-scale data loads. Depending on the nature of data and loading pipeline, ADWB automatically determines whether to use ETL or ELT mechanisms.

ADWB supports a large contingent of data sources including on-premises databases, cloud databases, files, ERP applications, CRM applications, and web services. You can build a loading process that is based on your data sources, overall architecture, and preferences. Here is a discussion of some data sources.

Staging database

If you are loading data from a staging database, you can use ADWB's data model source functionality. This feature enables you to quickly reverse engineer the staging database into a data model and deploy that model. Once deployed, this model can be used as a data source where you can select data from multiple tables and map it to fact or dimension tables. The advantage of this approach is that ADWB's Query Engine builds the most efficient query for you. The Query Engine can construct queries for data that is spread across multiple tables, creating automatic joins between these tables.



Reverse Engineering

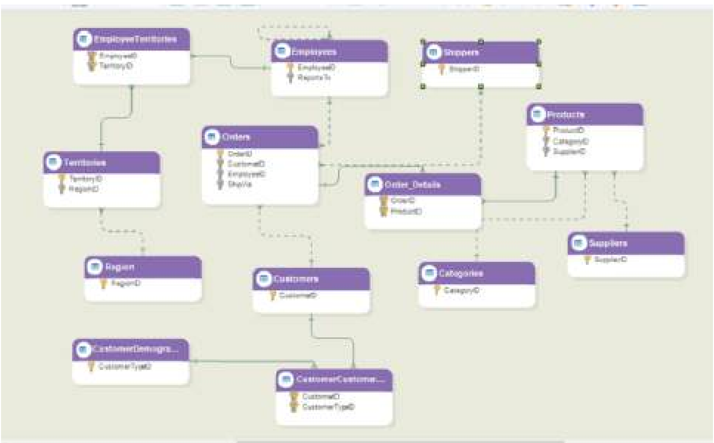
Database Connection

SQL Server: vmqa407; DB: NORTHWND. User: sa.

Entities

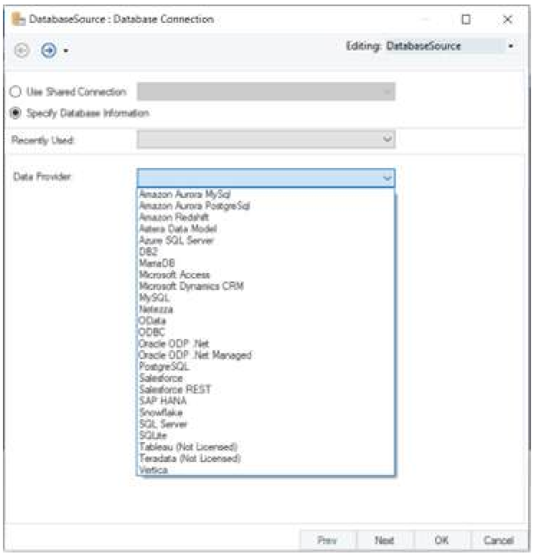
Search...

□	Existing	Type	Schema	Name
□	No	Table	dbo	BigTable
□	Yes	Table	dbo	Categories
□	Yes	Table	dbo	CustomerCustomerDemo
□	Yes	Table	dbo	CustomerDemographics
□	Yes	Table	dbo	Customers
□	Yes	Table	dbo	Employees
□	Yes	Table	dbo	EmployeeTerritories
□	Yes	Table	dbo	Order Details
□	Yes	Table	dbo	Orders
□	Yes	Table	dbo	Products
□	Yes	Table	dbo	Region
□	Yes	Table	dbo	Shippers
□	Yes	Table	dbo	Suppliers
□	Yes	Table	dbo	Territories



Database sources

ADWB supports almost all the popular cloud and on-premises databases. A list of supported databases is found here. To support incremental loading, change change-data-capture is also available for many of the supported databases.



DatabaseSource : Database Connection

Editing: DatabaseSource

○ Use Shared Connection

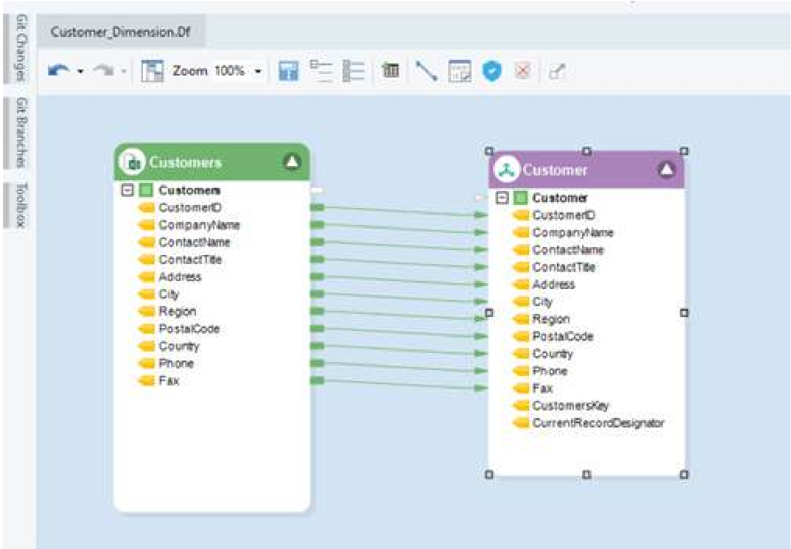
● Specify Database Information

CRM Systems

Astera provides native connectivity to popular CRM and ERP systems for data extraction. These include SAP, Microsoft Dynamics, Salesforce, and others. Other CRM systems can be accessed using our web-service based connectors.

Loading data from files

Common file formats such as delimited, fixed length, XML, JSON, Excel, and Parquet are supported. Astera also provides the ability to extract data from unstructured sources such as printed documents, PDF, images, voice data, and others.



Publishing a data warehouse

ADWB enables you to publish dimensional models. These models than can be accessed for query via Astera's data model source or OData. This enables users to access these data models from Business Intelligence tools such as Tableau, PowerBI, Qlik, and others.

Security

Users can also define permissions on the deployed data models to control access to these models. This enables you to create data marts for specific users, groups, or departments where they only have access to the tables that are part of those marts.

Publishing data marts

In addition to publishing the entire data warehouse, you can also publish subsets of data models—subject areas—as individual data marts. These data marts enable you to restrict access to data for specific users, groups, or departments.

Living in a hybrid world

Most organizations these days have a hybrid model. They have data residing in a combination of cloud and on-premises systems. Often, on-premises systems are behind a firewall and cannot be directly accessed from the cloud. If you are building a data warehouse in the cloud, you need secure mechanism to be able to retrieve data from on-premises systems. This is where Astera Data Services can help you create a secure point-to-point connectivity.

Conversely, you may need to access data in cloud files or other cloud data sources. Astera's technologies provide you with easy to use, secure, and seamless connectivity to data in all these scenarios.

Conclusion

Astera Data Warehouse Builder is a key component of Astera Data Stack (ADS). ADS is a constellation of many data management functionalities in a unified platform. These include high-performance ETL/ELT, unstructured data extraction, data preparation, data analytics, data services, and more. For more information about ADS, see here.

Whether you are building a data warehouse from the ground up or modernizing an existing warehouse, ADWB and Astera's broader offerings provide you will all the tools and technologies to assist you in rapidly building and deploying a data warehouse.

Astera
Enabling **Data-Driven** Innovation

www.astera.com Contact us for more information or to request a free trial
sales@astera.com | 888-77-ASTERA