

Lite Edition

# Data Dictionary

## Data Dictionary

28/04/2020 00:00:00





## Table of contents

1.	AdventureWorks database .....	10
1.1.	Business Entities .....	11
1.2.	People .....	12
1.2.1.	Tables .....	12
1.2.1.1.	Table: Person.Address .....	12
1.2.1.2.	Table: Person.AddressType .....	13
1.2.1.3.	Table: Person.BusinessEntity .....	14
1.2.1.4.	Table: Person.BusinessEntityAddress .....	14
1.2.1.5.	Table: Person.BusinessEntityContact .....	15
1.2.1.6.	Table: Person.ContactType .....	16
1.2.1.7.	Table: Person.CountryRegion .....	16
1.2.1.8.	Table: Person.EmailAddress .....	17
1.2.1.9.	Table: Person.Password .....	18
1.2.1.10.	Table: Person.Person .....	18
1.2.1.11.	Table: Person.PersonPhone .....	20
1.2.1.12.	Table: Person.PhoneNumberType .....	20
1.2.1.13.	Table: Person.StateProvince .....	21
1.2.2.	Views .....	22
1.2.2.1.	View: Person.vAdditionalContactInfo .....	22
1.2.2.2.	View: Person.vStateProvinceCountryRegion .....	22
1.3.	Human Resources .....	24
1.3.1.	Tables .....	24
1.3.1.1.	Table: HumanResources.Department .....	24
1.3.1.2.	Table: HumanResources.Employee .....	25
1.3.1.3.	Table: HumanResources.EmployeeDepartmentHistory .....	26
1.3.1.4.	Table: HumanResources.EmployeePayHistory .....	27
1.3.1.5.	Table: HumanResources.JobCandidate .....	28
1.3.1.6.	Table: HumanResources.Shift .....	28
1.3.2.	Views .....	29
1.3.2.1.	View: HumanResources.vEmployee .....	29
1.3.2.2.	View: HumanResources.vEmployeeDepartment .....	29
1.3.2.3.	View: HumanResources.vEmployeeDepartmentHistory .....	30
1.3.2.4.	View: HumanResources.vJobCandidate .....	30
1.3.2.5.	View: HumanResources.vJobCandidateEducation .....	31
1.3.2.6.	View: HumanResources.vJobCandidateEmployment .....	31
1.3.3.	Procedures .....	32
1.3.3.1.	Procedure: dbo.uspGetEmployeeManagers .....	32
1.3.3.2.	Procedure: dbo.uspGetManagerEmployees .....	32
1.3.3.3.	Procedure: dbo.uspSearchCandidateResumes .....	32
1.3.3.4.	Procedure: HumanResources.uspUpdateEmployeeHireInfo .....	32
1.3.3.5.	Procedure: HumanResources.uspUpdateEmployeeLogin .....	33
1.3.3.6.	Procedure: HumanResources.uspUpdateEmployeePersonalInfo .....	33
1.3.4.	Functions .....	33
1.3.4.1.	Function: dbo.ufnGetContactInformation .....	33
1.4.	Products .....	34

1.4.1. Tables .....	34
1.4.1.1. Table: Production.Culture .....	34
1.4.1.2. Table: Production.Document .....	35
1.4.1.3. Table: Production.Illustration .....	36
1.4.1.4. Table: Production.Product .....	36
1.4.1.5. Table: Production.ProductCategory .....	39
1.4.1.6. Table: Production.ProductDescription .....	39
1.4.1.7. Table: Production.ProductDocument .....	40
1.4.1.8. Table: Production.ProductModel .....	40
1.4.1.9. Table: Production.ProductModelIllustration .....	41
1.4.1.10. Table: Production.ProductModelProductDescriptionCulture .....	42
1.4.1.11. Table: Production.ProductPhoto .....	42
1.4.1.12. Table: Production.ProductProductPhoto .....	43
1.4.1.13. Table: Production.ProductReview .....	44
1.4.1.14. Table: Production.ProductSubcategory .....	44
1.4.1.15. Table: Production.UnitMeasure .....	45
1.4.2. Views .....	46
1.4.2.1. View: Production.vProductAndDescription .....	46
1.4.2.2. View: Production.vProductModelCatalogDescription .....	46
1.4.2.3. View: Production.vProductModellInstructions .....	47
1.4.3. Procedures .....	47
1.4.3.1. Procedure: dbo.uspGetBillOfMaterials .....	47
1.4.3.2. Procedure: dbo.uspGetWhereUsedProductID .....	48
1.4.4. Functions .....	48
1.4.4.1. Function: dbo.ufnGetDocumentStatusText .....	48
1.4.4.2. Function: dbo.ufnGetProductDealerPrice .....	48
1.4.4.3. Function: dbo.ufnGetProductListPrice .....	48
1.4.4.4. Function: dbo.ufnGetProductStandardCost .....	48
1.4.4.5. Function: dbo.ufnGetStock .....	49
1.5. Manufacturing .....	50
1.5.1. Tables .....	50
1.5.1.1. Table: Production.BillOfMaterials .....	50
1.5.1.2. Table: Production.ProductCostHistory .....	51
1.5.1.3. Table: Production.ProductListPriceHistory .....	52
1.5.1.4. Table: Production.ScrapReason .....	52
1.5.1.5. Table: Production.TransactionHistory .....	53
1.5.1.6. Table: Production.TransactionHistoryArchive .....	53
1.5.1.7. Table: Production.WorkOrder .....	54
1.5.1.8. Table: Production.WorkOrderRouting .....	55
1.6. Purchasing .....	57
1.6.1. Tables .....	57
1.6.1.1. Table: Purchasing.ProductVendor .....	57
1.6.1.2. Table: Purchasing.PurchaseOrderDetail .....	58
1.6.1.3. Table: Purchasing.PurchaseOrderHeader .....	59
1.6.1.4. Table: Purchasing.ShipMethod .....	60
1.6.1.5. Table: Purchasing.Vendor .....	61

1.6.2. Views .....	62
1.6.2.1. View: Purchasing.vVendorWithAddresses .....	62
1.6.2.2. View: Purchasing.vVendorWithContacts .....	63
1.6.3. Functions .....	63
1.6.3.1. Function: dbo.ufnGetPurchaseOrderStatusText .....	63
1.7. Inventory .....	64
1.7.1. Tables .....	64
1.7.1.1. Table: Production.Location .....	64
1.7.1.2. Table: Production.ProductInventory .....	65
1.8. Sales .....	66
1.8.1. Tables .....	66
1.8.1.1. Table: Sales.CountryRegionCurrency .....	66
1.8.1.2. Table: Sales.CreditCard .....	67
1.8.1.3. Table: Sales.Currency .....	67
1.8.1.4. Table: Sales.CurrencyRate .....	68
1.8.1.5. Table: Sales.Customer .....	69
1.8.1.6. Table: Sales.PersonCreditCard .....	70
1.8.1.7. Table: Sales.SalesOrderDetail .....	70
1.8.1.8. Table: Sales.SalesOrderHeader .....	71
1.8.1.9. Table: Sales.SalesOrderHeaderSalesReason .....	74
1.8.1.10. Table: Sales.SalesPerson .....	74
1.8.1.11. Table: Sales.SalesPersonQuotaHistory .....	75
1.8.1.12. Table: Sales.SalesReason .....	76
1.8.1.13. Table: Sales.SalesTaxRate .....	76
1.8.1.14. Table: Sales.SalesTerritory .....	77
1.8.1.15. Table: Sales.SalesTerritoryHistory .....	78
1.8.1.16. Table: Sales.ShoppingCartItem .....	79
1.8.1.17. Table: Sales.SpecialOffer .....	80
1.8.1.18. Table: Sales.SpecialOfferProduct .....	80
1.8.1.19. Table: Sales.Store .....	81
1.8.2. Views .....	82
1.8.2.1. View: Sales.vIndividualCustomer .....	82
1.8.2.2. View: Sales.vPersonDemographics .....	82
1.8.2.3. View: Sales.vSalesPerson .....	83
1.8.2.4. View: Sales.vSalesPersonSalesByFiscalYears .....	84
1.8.2.5. View: Sales.vStoreWithAddresses .....	84
1.8.2.6. View: Sales.vStoreWithContacts .....	84
1.8.2.7. View: Sales.vStoreWithDemographics .....	85
1.8.3. Functions .....	85
1.8.3.1. Function: dbo.ufnGetAccountingEndDate .....	85
1.8.3.2. Function: dbo.ufnGetAccountingStartDate .....	85
1.8.3.3. Function: dbo.ufnGetSalesOrderStatusText .....	86
1.8.3.4. Function: dbo.ufnLeadingZeros .....	86
1.9. Admin .....	87
1.9.1. Tables .....	87
1.9.1.1. Table: dbo.AWBuildVersion .....	87

1.9.1.2. Table: dbo.DatabaseLog .....	87
1.9.1.3. Table: dbo.ErrorLog .....	87
1.9.2. Procedures .....	88
1.9.2.1. Procedure: dbo.usp.LogError .....	88
1.9.2.2. Procedure: dbo.uspPrintError .....	88
2. Dataedo repository .....	89
2.1. Schema and Metadata .....	90
2.1.1. Tables .....	91
2.1.1.1. Table: dbo.columns (Columns) .....	91
2.1.1.2. Table: dbo.databases (Documentations) .....	98
2.1.1.3. Table: dbo.modules (Modules) .....	104
2.1.1.4. Table: dbo.parameters (Parameters) .....	110
2.1.1.5. Table: dbo.procedures (Procedures and Functions) .....	116
2.1.1.6. Table: dbo.procedures_modules (Procedures-modules) .....	122
2.1.1.7. Table: dbo.tables (Tables and Views) .....	123
2.1.1.8. Table: dbo.tables_modules (Tables-modules) .....	130
2.1.1.9. Table: dbo.tables_relations (Relationships) .....	131
2.1.1.10. Table: dbo.tables_relations_columns (Relationships columns) .....	136
2.1.1.11. Table: dbo.triggers (Triggers) .....	138
2.1.1.12. Table: dbo.unique_constraints (Unique keys) .....	143
2.1.1.13. Table: dbo.unique_constraints_columns (Unique keys columns) .....	149
2.1.1.14. Table: dbo.user_connections (User connection details) .....	150
2.2. Schema Change Tracking .....	153
2.2.1. Tables .....	154
2.2.1.1. Table: dbo.columns_changes (Columns - Schema changes) .....	154
2.2.1.2. Table: dbo.parameters_changes (Parameters - Schema changes) .....	156
2.2.1.3. Table: dbo.procedures_changes (Procedures and Functions - Schema changes) .....	157
2.2.1.4. Table: dbo.schema_updates (Imports/Updates) .....	158
2.2.1.5. Table: dbo.tables_changes (Tables and Views - Schema changes) .....	160
2.2.1.6. Table: dbo.tables_relations_changes (Relationships - Schema changes) .....	162
2.2.1.7. Table: dbo.tables_relations_columns_changes (Relationships columns - Schema changes) .....	164
2.2.1.8. Table: dbo.triggers_changes (Triggers - Schema changes) .....	165
2.2.1.9. Table: dbo.unique_constraints_changes (Unique keys - Schema changes) .....	167
2.2.1.10. Table: dbo.unique_constraints_columns_changes (Unique keys columns - Schema changes) .....	168
2.3. ERDs .....	170
2.3.1. Tables .....	171
2.3.1.1. Table: dbo.erd_links (ERD links) .....	171
2.3.1.2. Table: dbo.erd_nodes (ERD entities) .....	172
2.3.1.3. Table: dbo.erd_nodes_columns (ERD entities columns) .....	173
2.4. Business Glossary .....	175
2.4.1. Tables .....	175
2.4.1.1. Table: dbo.glossary_mappings (Term-Data links) .....	175
2.4.1.2. Table: dbo.glossary_term_relationship_types (Term Relationship Types) .....	176
2.4.1.3. Table: dbo.glossary_term_relationships (Term Relationships) .....	177
2.4.1.4. Table: dbo.glossary_term_types (Term Types) .....	178
2.4.1.5. Table: dbo.glossary_terms (Business Glossary Terms) .....	179

2.5. Data Classification .....	185
2.5.1. Tables .....	185
2.5.1.1. Table: dbo.classifier_masks (Data Classification Column Masks) .....	185
2.5.1.2. Table: dbo.classifier_rules (Data Classification Domain Rules) .....	186
2.5.1.3. Table: dbo.classifiers (Data Classification domain) .....	188
2.5.1.4. Table: virtual.classifier_data_domains (Data Classification Data Domains) .....	190
2.5.1.5. Table: virtual.column_name_mask_match .....	191
2.6. Dependencies .....	192
2.6.1. Tables .....	192
2.6.1.1. Table: dbo.dependencies (Dependencies) .....	192
2.6.1.2. Table: dbo.dependencies_descriptions (Dependencies descriptions) .....	193
2.7. Custom fields .....	195
2.7.1. Tables .....	196
2.7.1.1. Table: dbo.custom_fields (Custom fields) .....	196
2.7.1.2. Table: dbo.custom_fields_values (Custom fields index) .....	202
2.7.1.3. Table: dbo.documentation_custom_fields (Documentation custom fields) .....	203
2.8. Utils .....	205
2.8.1. Tables .....	205
2.8.1.1. Table: dbo.guid (Repo ID) .....	205
2.8.1.2. Table: dboignored_objects (Ignored objects) .....	205
2.8.1.3. Table: dbo.licenses (Users) .....	206
2.8.1.4. Table: dbo.version (Repository version) .....	207
2.9. Other .....	209
2.9.1. Tables .....	209
2.9.1.1. Table: dbo.custom_field_classes (Field class) .....	209

## Legend

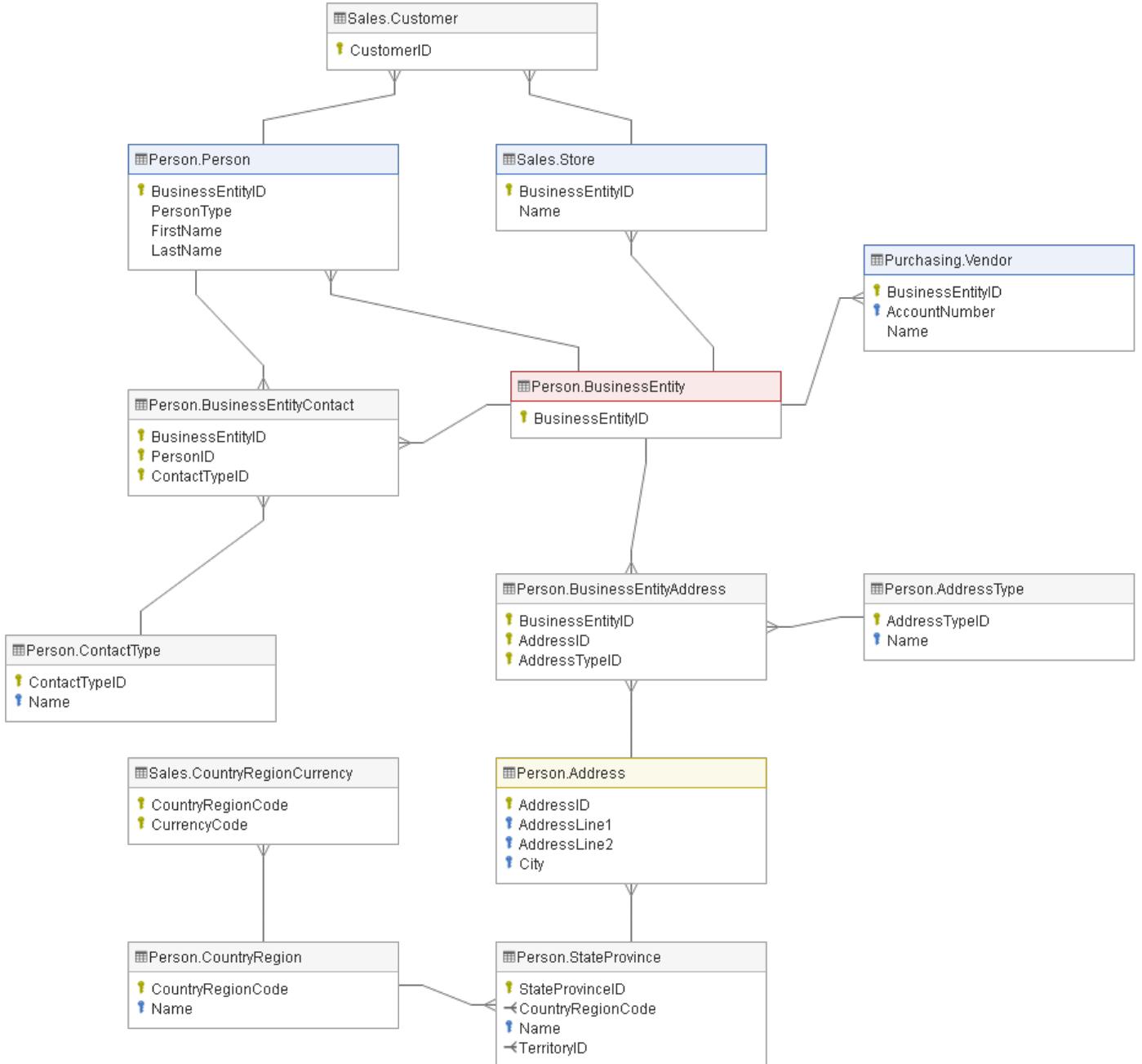
- 🔑 Primary key
- 🔑 Primary key disabled
- 🔑 User-defined primary key
- 🔑 Unique key
- 🔑 Unique key disabled
- 🔑 User-defined unique key
- ⚡ Active trigger
- ⚡ Disabled trigger
- Many to one relation
- User-defined many to one relation
- ← One to many relation
- ← User-defined one to many relation
- One to one relation
- User-defined one to one relation
- @ Input
- @→ Output
- \*@ Input/Output
- ⬆️ Uses dependency
- ⬆️ User-defined uses dependency
- ⬇️ Used by dependency
- ⬇️ User-defined used by dependency

## 1. AdventureWorks database

**AdventureWorks** is a sample database for Microsoft SQL Server 2008 to 2014. This is a documentation of this database created with [Dataedo](#).

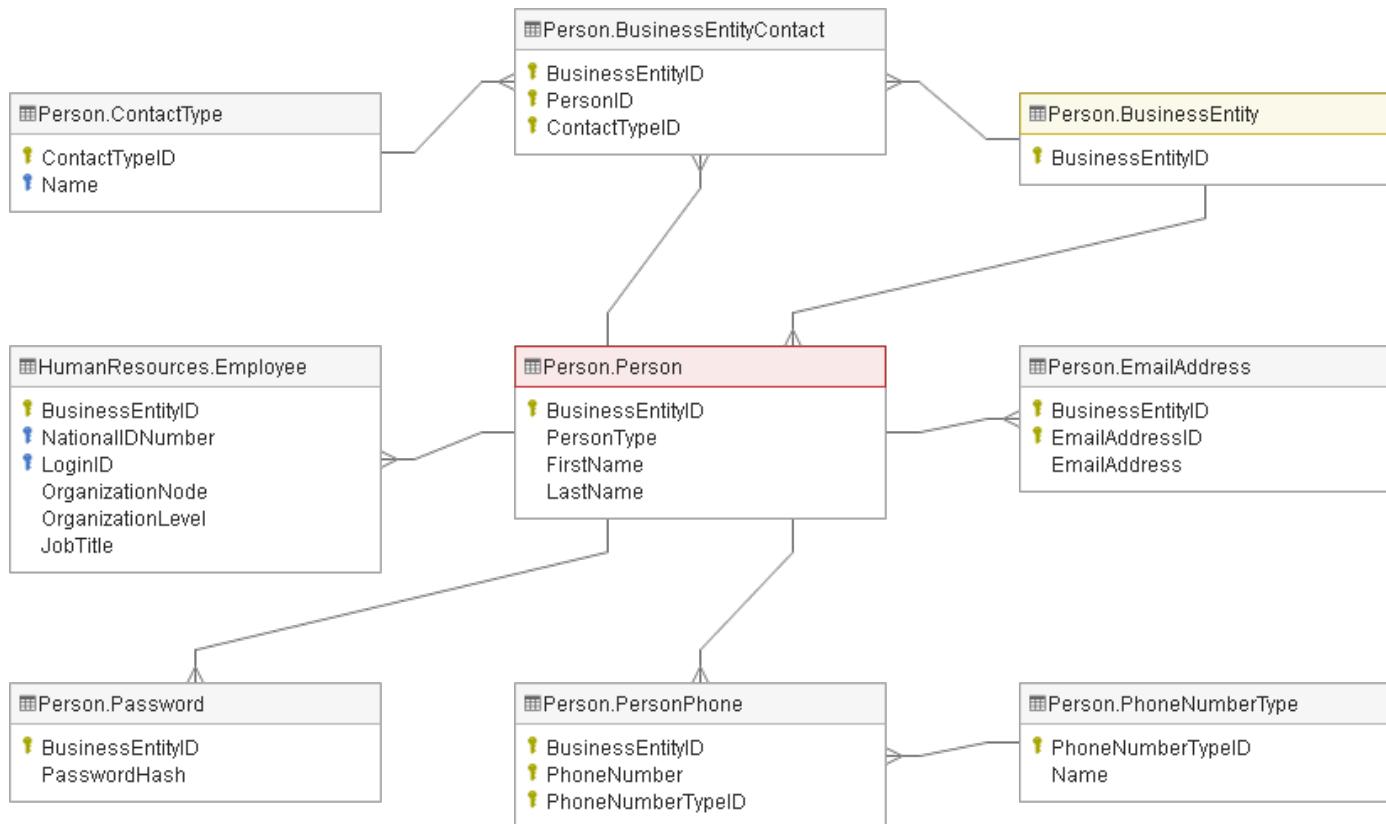
AdventureWorks database supports standard online transaction processing scenarios for a fictitious bicycle manufacturer - **Adventure Works Cycles**. Scenarios include Manufacturing, Sales, Purchasing, Product Management, Contact Management, and Human Resources.

## 1.1. Business Entities



Vendors, customers, and employees have common tables for addresses and contacts. Those tables are linked to one table - **BusinessEntity** that holds ID for all vendors, customers, and employees tables.

## 1.2. People



Names and addresses of individual customers, vendors, and employees.

### 1.2.1. Tables

### 1.2.1.1. Table: Person.Address

Street address information for customers, employees, and vendors.

## Columns

	Name	Data type	Description / Attributes
	AddressID	int	Primary key for Address records. <b>Identity / Auto increment</b>
	AddressLine1	nvarchar(60)	First street address line.
	AddressLine2	nvarchar(60)	Second street address line. <b>Nullable</b>
	City	nvarchar(30)	Name of the city.
	StateProvinceID	int	Unique identification number for the state or province. Foreign key to StateProvince table. <b>References:</b> Person.StateProvince
	PostalCode	nvarchar(15)	Postal code for the street address.
	SpatialLocation	geography	Latitude and longitude of this address. <b>Nullable</b>
	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Person.StateProvince	<b>Person.Address.StateProvinceID = Person.StateProvince.StateProvinceID</b>	FK_Address_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.StateProvinceID.

## Linked from

	Table	Join	Title / Name / Description
←	Person.BusinessEntityAddress	<b>Person.Address.AddressID = Person.BusinessEntityAddress.AddressID</b>	FK_BusinessEntityAddress_Address_AddressID Foreign key constraint referencing Address.AddressID.
←	Sales.SalesOrderHeader	<b>Person.Address.AddressID = Sales.SalesOrderHeader.BillToAddressID</b>	FK_SalesOrderHeader_Address_BillToAddressID Foreign key constraint referencing Address.AddressID.
←	Sales.SalesOrderHeader	<b>Person.Address.AddressID = Sales.SalesOrderHeader.ShipToAddressID</b>	FK_SalesOrderHeader_Address_ShipToAddressID Foreign key constraint referencing Address.AddressID.

## Unique keys

	Columns	Name / Description
🔑	AddressID	PK_Address_AddressID Primary key (clustered) constraint
🔑	rowguid	AK_Address_rowguid Unique nonclustered index. Used to support replication samples.
🔑	AddressLine1, AddressLine2, City, StateProvinceID, PostalCode	IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode Nonclustered index.

## 1.2.1.2. Table: Person.AddressType

Types of addresses stored in the Address table.

### Columns

	Name	Data type	Description / Attributes
🔑	AddressTypeID	int	Primary key for AddressType records. <b>Identity / Auto increment</b>
🔑	Name	nvarchar(50)	Address type description. For example, Billing, Home, or Shipping.
🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
⌚	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
←	Person.BusinessEntityAddress	<b>Person.AddressType.AddressTypeID = Person.BusinessEntityAddress.AddressTypeID</b>	FK_BusinessEntityAddress_AddressType_AddressTypeID Foreign key constraint referencing AddressType.AddressTypeID.

## Unique keys

	Columns	Name / Description
🔑	AddressTypeID	PK_AddressType_AddressTypeID Primary key (clustered) constraint

	Columns	Name / Description
🔑	Name	AK_AddressType_Name Unique nonclustered index.
🔑	rowguid	AK_AddressType_rowguid Unique nonclustered index. Used to support replication samples.

### 1.2.1.3. Table: Person.BusinessEntity

Source of the ID that connects vendors, customers, and employees with address and contact information.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Primary key for all customers, vendors, and employees. <b>Identity / Auto increment</b>
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Linked from

	Table	Join	Title / Name / Description
→	Person.BusinessEntityAddress	<b>Person.BusinessEntity.BusinessEntityID</b> = Person.BusinessEntityAddress.BusinessEntityID	FK_BusinessEntityAddress_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.
→	Person.BusinessEntityContact	<b>Person.BusinessEntity.BusinessEntityID</b> = Person.BusinessEntityContact.BusinessEntityID	FK_BusinessEntityContact_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.
→	Person.Person	<b>Person.BusinessEntity.BusinessEntityID</b> = Person.Person.BusinessEntityID	FK_Person_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.
→	Sales.Store	<b>Person.BusinessEntity.BusinessEntityID</b> = Sales.Store.BusinessEntityID	FK_Store_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID
→	Purchasing.Vendor	<b>Person.BusinessEntity.BusinessEntityID</b> = Purchasing.Vendor.BusinessEntityID	FK_Vendor_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID

#### Unique keys

	Columns	Name / Description
🔑	BusinessEntityID	PK_BusinessEntity_BusinessEntityID Primary key (clustered) constraint
🔑	rowguid	AK_BusinessEntity_rowguid Unique nonclustered index. Used to support replication samples.

### 1.2.1.4. Table: Person.BusinessEntityAddress

Cross-reference table mapping customers, vendors, and employees to their addresses.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Primary key. Foreign key to BusinessEntity.BusinessEntityID. <b>References:</b> Person.BusinessEntity

		Name	Data type	Description / Attributes
█	█	AddressID	int	Primary key. Foreign key to Address.AddressID. <b>References:</b> Person.Address
█	█	AddressTypeID	int	Primary key. Foreign key to AddressType.AddressTypeID. <b>References:</b> Person.AddressType
█	█	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.Address	<b>Person.BusinessEntityAddress</b> .AddressID = Person.Address.AddressID	FK_BusinessEntityAddress_Address_AddressID Foreign key constraint referencing Address.AddressID.
→	Person.AddressType	<b>Person.BusinessEntityAddress</b> .AddressTypeID = Person.AddressType.AddressTypeID	FK_BusinessEntityAddress_AddressType_AddressTypeID Foreign key constraint referencing AddressType.AddressTypeID.
→	Person.BusinessEntity	<b>Person.BusinessEntityAddress</b> .BusinessEntityID = Person.BusinessEntity.BusinessEntityID	FK_BusinessEntityAddress_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.

Unique keys

	Columns	Name / Description
█	BusinessEntityID, AddressID, AddressTypeID	PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID Primary key (clustered) constraint
█	rowguid	AK_BusinessEntityAddress_rowguid Unique nonclustered index. Used to support replication samples.

### 1.2.1.5. Table: Person.BusinessEntityContact

Cross-reference table mapping stores, vendors, and employees to people

Columns

		Name	Data type	Description / Attributes
█	█	BusinessEntityID	int	Primary key. Foreign key to BusinessEntity.BusinessEntityID. <b>References:</b> Person.BusinessEntity
█	█	PersonID	int	Primary key. Foreign key to Person.BusinessEntityID. <b>References:</b> Person.Person
█	█	ContactTypeID	int	Primary key. Foreign key to ContactType.ContactTypeID. <b>References:</b> Person.ContactType
█	█	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.BusinessEntity	<b>Person.BusinessEntityContact</b> .BusinessEntityID = Person.BusinessEntity.BusinessEntityID	FK_BusinessEntityContact_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.

	Table	Join	Title / Name / Description
→	Person.ContactType	<b>Person.BusinessEntityContact.ContactTypeID</b> = Person.ContactType.ContactTypeID	FK_BusinessEntityContact_ContactType_ContactTypeID Foreign key constraint referencing ContactType.ContactTypeID.
→	Person.Person	<b>Person.BusinessEntityContact.PersonID</b> = Person.Person.BusinessEntityID	FK_BusinessEntityContact_Person_PersonID Foreign key constraint referencing Person.BusinessEntityID.

## Unique keys

	Columns	Name / Description
🔑	BusinessEntityID, PersonID, ContactTypeID	PK_BusinessEntityContact_BusinessEntityID_PersonID_ContactTypeID Primary key (clustered) constraint
🔑	rowguid	AK_BusinessEntityContact_rowguid Unique nonclustered index. Used to support replication samples.

### 1.2.1.6. Table: Person.ContactType

Lookup table containing the types of business entity contacts.

#### Columns

	Name	Data type	Description / Attributes
☰	>ContactTypeID	int	Primary key for ContactType records. <b>Identity / Auto increment</b>
☰	Name	nvarchar(50)	Contact type description.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Linked from

	Table	Join	Title / Name / Description
←	Person.BusinessEntityContact	<b>Person.ContactType.ContactTypeID</b> = Person.BusinessEntityContact.ContactTypeID	FK_BusinessEntityContact_ContactType_ContactTypeID Foreign key constraint referencing ContactType.ContactTypeID.

## Unique keys

	Columns	Name / Description
🔑	ContactTypeID	PK_ContactType_ContactTypeID Primary key (clustered) constraint
🔑	Name	AK_ContactType_Name Unique nonclustered index.

### 1.2.1.7. Table: Person.CountryRegion

Lookup table containing the ISO standard codes for countries and regions.

#### Columns

	Name	Data type	Description / Attributes
☰	CountryRegionCode	nvarchar(3)	ISO standard code for countries and regions.
☰	Name	nvarchar(50)	Country or region name.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Sales.CountryRegionCurrency	<b>Person.CountryRegion.CountryRegion</b> Code = Sales.CountryRegionCurrency.CountryRegionCode	FK_CountryRegionCurrency_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.
→	Sales.SalesTerritory	<b>Person.CountryRegion.CountryRegion</b> Code = Sales.SalesTerritory.CountryRegionCode	FK_SalesTerritory_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.
→	Person.StateProvince	<b>Person.CountryRegion.CountryRegion</b> Code = Person.StateProvince.CountryRegionCode	FK_StateProvince_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.

## Unique keys

	Columns	Name / Description
🔑	CountryRegionCode	PK_CountryRegion_CountryRegionCode Primary key (clustered) constraint
🔑	Name	AK_CountryRegion_Name Unique nonclustered index.

## 1.2.1.8. Table: Person.EmailAddress

Where to send a person email.

### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Primary key. Person associated with this email address. Foreign key to Person.BusinessEntityID <b>References:</b> Person.Person
☰	EmailAddressID	int	Primary key. ID of this email address. <b>Identity / Auto increment</b>
☰	EmailAddress	nvarchar(50)	E-mail address for the person. <b>Nullable</b>
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Person.Person	<b>Person.EmailAddress.BusinessEntityID</b> = Person.Person.BusinessEntityID	FK_EmailAddress_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.

## Unique keys

	Columns	Name / Description
🔑	BusinessEntityID, EmailAddressID	PK_EmailAddress_BusinessEntityID_EmailAddressID Primary key (clustered) constraint

## 1.2.1.9. Table: Person.Password

One way hashed authentication information

Columns

	Name	Data type	Description / Attributes
🔑	BusinessEntityID	int	<b>References:</b> Person.Person
🔑	PasswordHash	varchar(128)	Password for the e-mail account.
🔑	PasswordSalt	varchar(10)	Random value concatenated with the password string before the password is hashed.
🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
🔑	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.Person	<b>Person.Password.BusinessEntityID</b> = Person.Person.BusinessEntityID	FK_Password_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.

Unique keys

	Columns	Name / Description
🔑	BusinessEntityID	PK_Password_BusinessEntityID Primary key (clustered) constraint

## 1.2.1.10. Table: Person.Person

Human beings involved with AdventureWorks: employees, customer contacts, and vendor contacts.

Columns

	Name	Data type	Description / Attributes
🔑	BusinessEntityID	int	Primary key for Person records. <b>References:</b> Person.BusinessEntity
🔑	PersonType	nchar(2)	Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-sales), VC = Vendor contact, GC = General contact
🔑	NameStyle	bit	0 = The data in FirstName and LastName are stored in western style (first name, last name) order. 1 = Eastern style (last name, first name) order. <b>Default:</b> 0
🔑	Title	nvarchar(8)	A courtesy title. For example, Mr. or Ms. <b>Nullable</b>
🔑	FirstName	nvarchar(50)	First name of the person.
🔑	MiddleName	nvarchar(50)	Middle name or middle initial of the person. <b>Nullable</b>
🔑	LastName	nvarchar(50)	Last name of the person.
🔑	Suffix	nvarchar(10)	Surname suffix. For example, Sr. or Jr. <b>Nullable</b>

	Name	Data type	Description / Attributes
>EmailPromotion	EmailPromotion	int	0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from AdventureWorks, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and selected partners. <b>Default:</b> 0
AdditionalContactInfo	AdditionalContactInfo	xml	Additional contact information about the person stored in xml format. <b>Nullable</b>
Demographics	Demographics	xml	Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis. <b>Nullable</b>
rowguid	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
ModifiedDate	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
Person.BusinessEntity	Person.BusinessEntity	<b>Person.Person.BusinessEntityID</b> = Person.BusinessEntity.BusinessEntityID	FK_Person_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID.

## Linked from

	Table	Join	Title / Name / Description
Person.BusinessEntityContact	Person.BusinessEntityContact	<b>Person.Person.BusinessEntityID</b> = Person.BusinessEntityContact.PersonID	FK_BusinessEntityContact_Person_PersonID Foreign key constraint referencing Person.BusinessEntityID.
Sales.Customer	Sales.Customer	<b>Person.Person.BusinessEntityID</b> = Sales.Customer.PersonID	FK_Customer_Person_PersonID Foreign key constraint referencing Person.BusinessEntityID.
Person.EmailAddress	Person.EmailAddress	<b>Person.Person.BusinessEntityID</b> = Person.EmailAddress.BusinessEntityID	FK_EmailAddress_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.
HumanResources.Employee	HumanResources.Employee	<b>Person.Person.BusinessEntityID</b> = HumanResources.Employee.BusinessEntityID	FK_Employee_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.
Person.Password	Person.Password	<b>Person.Person.BusinessEntityID</b> = Person.Password.BusinessEntityID	FK_Password_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.
Sales.PersonCreditCard	Sales.PersonCreditCard	<b>Person.Person.BusinessEntityID</b> = Sales.PersonCreditCard.BusinessEntityID	FK_PersonCreditCard_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.
Person.PersonPhone	Person.PersonPhone	<b>Person.Person.BusinessEntityID</b> = Person.PersonPhone.BusinessEntityID	FK_PersonPhone_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.

## Unique keys

	Columns	Name / Description
BusinessEntityID	BusinessEntityID	PK_Person_BusinessEntityID Primary key (clustered) constraint
rowguid	rowguid	AK_Person_rowguid Unique nonclustered index. Used to support replication samples.

## Triggers

	Name	When	Description
⚡	iuPerson	After Insert, Update	AFTER INSERT, UPDATE trigger inserting Individual only if the Customer does not exist in the Store table and setting the ModifiedDate column in the Person table to the current date.

### 1.2.1.11. Table: Person.PersonPhone

Telephone number and type of a person.

#### Columns

	Name	Data type	Description / Attributes
🔑	BusinessEntityID	int	Business entity identification number. Foreign key to Person.BusinessEntityID. <b>References:</b> Person.Person
🔑	PhoneNumber	nvarchar(25)	Telephone number identification number.
🔑	PhoneNumberTypeID	int	Kind of phone number. Foreign key to PhoneNumberType.PhoneNumberTypeID. <b>References:</b> Person.PhoneNumberType
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Links to

	Table	Join	Title / Name / Description
→	Person.Person	<b>Person.PersonPhone.BusinessEntityID</b> = Person.Person.BusinessEntityID	FK_PersonPhone_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.
→	Person.PhoneNumberType	<b>Person.PersonPhone.PhoneNumberTypeID</b> = Person.PhoneNumberType.PhoneNumberTypeID	FK_PersonPhone_PhoneNumberType_PhoneNumberTypeID Foreign key constraint referencing PhoneNumberType.PhoneNumberTypeID.

#### Unique keys

	Columns	Name / Description
🔑	BusinessEntityID, PhoneNumber, PhoneNumberTypeID	PK_PersonPhone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID Primary key (clustered) constraint

### 1.2.1.12. Table: Person.PhoneNumberType

Type of phone number of a person.

#### Columns

	Name	Data type	Description / Attributes
🔑	PhoneNumberTypeID	int	Primary key for telephone number type records. <b>Identity / Auto increment</b>
🔑	Name	nvarchar(50)	Name of the telephone number type
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Person.PersonPhone	<b>Person.PhoneNumberType.PhoneNumberTypeID</b> = Person.PersonPhone.PhoneNumberTypeID Person.PersonPhone.PhoneNumberTypeID	FK_PersonPhone_PhoneNumberType_PhoneNumberTypeID Foreign key constraint referencing PhoneNumberType.PhoneNumberTypeID.

## Unique keys

	Columns	Name / Description
🔑	PhoneNumberTypeID	PK_PhoneNumberType_PhoneNumberTypeID Primary key (clustered) constraint

## 1.2.1.13. Table: Person.StateProvince

State and province lookup table.

### Columns

	Name	Data type	Description / Attributes
primaryKey	StateProvinceID	int	Primary key for StateProvince records. <b>Identity / Auto increment</b>
primaryKey	StateProvinceCode	nchar(3)	ISO standard state or province code.
primaryKey	CountryRegionCode	nvarchar(3)	ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode. <b>References:</b> Person.CountryRegion
primaryKey	IsOnlyStateProvinceFlag	bit	0 = StateProvinceCode exists. 1 = StateProvinceCode unavailable, using CountryRegionCode. <b>Default:</b> 1
primaryKey	Name	nvarchar(50)	State or province description.
primaryKey	TerritoryID	int	ID of the territory in which the state or province is located. Foreign key to SalesTerritory.SalesTerritoryID. <b>References:</b> Sales.SalesTerritory
primaryKey	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
primaryKey	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Person.CountryRegion	<b>Person.StateProvince.CountryRegionCode</b> = Person.CountryRegion.CountryRegionCode	FK_StateProvince_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.
→	Sales.SalesTerritory	<b>Person.StateProvince.TerritoryID</b> = Sales.SalesTerritory.TerritoryID	FK_StateProvince_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.

## Linked from

	Table	Join	Title / Name / Description
→	Person.Address	<b>Person.StateProvince.StateProvinceID</b> = Person.Address.StateProvinceID	FK_Address_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.StateProvinceID.
→	Sales.SalesTaxRate	<b>Person.StateProvince.StateProvinceID</b> = Sales.SalesTaxRate.StateProvinceID	FK_SalesTaxRate_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.StateProvinceID.

## Unique keys

	Columns	Name / Description
🔑	StateProvinceID	PK_StateProvince_StateProvinceID Primary key (clustered) constraint
🔑	Name	AK_StateProvince_Name Unique nonclustered index.
🔑	rowguid	AK_StateProvince_rowguid Unique nonclustered index. Used to support replication samples.
🔑	StateProvinceCode, CountryRegionCode	AK_StateProvince_StateProvinceCode_CountryRegionCode Unique nonclustered index.

## 1.2.2. Views

### 1.2.2.1. View: Person.vAdditionalContactInfo

Displays the contact name and content from each element in the xml column AdditionalContactInfo for that person.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	
☰	FirstName	nvarchar(50)	
☰	MiddleName	nvarchar(50)	<b>Nullable</b>
☰	LastName	nvarchar(50)	
☰	TelephoneNumber	nvarchar(50)	<b>Nullable</b>
☰	TelephoneSpecialInstructions	nvarchar(MAX)	<b>Nullable</b>
☰	Street	nvarchar(50)	<b>Nullable</b>
☰	City	nvarchar(50)	<b>Nullable</b>
☰	StateProvince	nvarchar(50)	<b>Nullable</b>
☰	PostalCode	nvarchar(50)	<b>Nullable</b>
☰	CountryRegion	nvarchar(50)	<b>Nullable</b>
☰	HomeAddressSpecialInstructions	nvarchar(MAX)	<b>Nullable</b>
☰	EMailAddress	nvarchar(128)	<b>Nullable</b>
☰	EMailSpecialInstructions	nvarchar(MAX)	<b>Nullable</b>
☰	EMailTelephoneNumber	nvarchar(50)	<b>Nullable</b>
☰	rowguid	uniqueidentifier	
☰	ModifiedDate	datetime	

### 1.2.2.2. View: Person.vStateProvinceCountryRegion

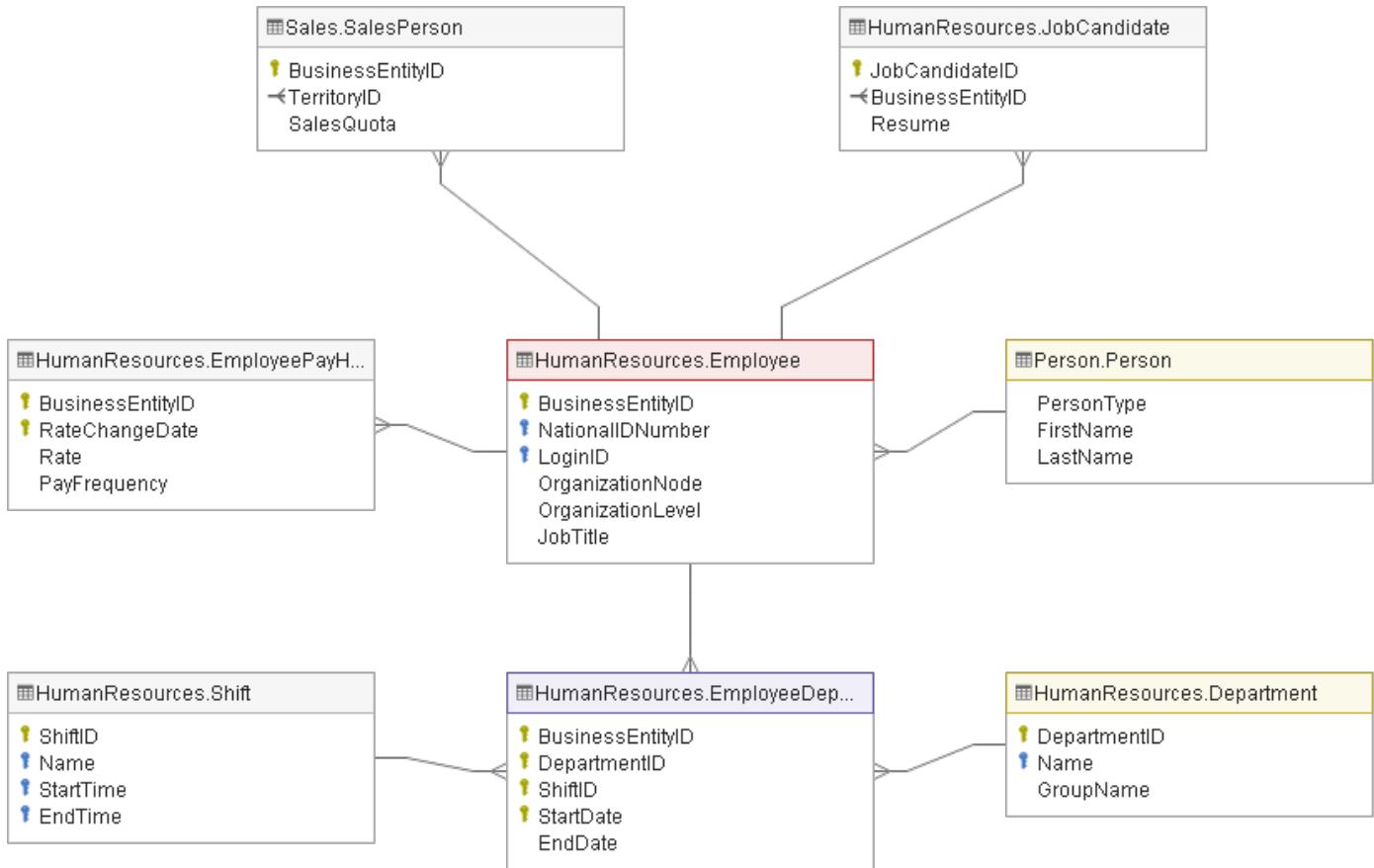
Joins StateProvince table with CountryRegion table.

#### Columns

	Name	Data type	Description / Attributes
☰	StateProvinceID	int	
☰	StateProvinceCode	nchar(3)	
☰	IsOnlyStateProvinceFlag	bit	

	Name	Data type	Description / Attributes
■	StateProvinceName	nvarchar(50)	
■	TerritoryID	int	
■	CountryRegionCode	nvarchar(3)	
■	CountryRegionName	nvarchar(50)	

## 1.3. Human Resources



Employees of Adventure Works Cycles.

### 1.3.1. Tables

#### 1.3.1.1. Table: HumanResources.Department

Lookup table containing the departments within the Adventure Works Cycles company.

##### Columns

	Name	Data type	Description / Attributes
PK	DepartmentID	smallint	Primary key for Department records. <b>Identity / Auto increment</b>
PK	Name	nvarchar(50)	Name of the department.
	GroupName	nvarchar(50)	Name of the group to which the department belongs.
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

##### Linked from

	Table	Join	Title / Name / Description
FK	HumanResources.EmployeeDepartmentHistory	<b>HumanResources.Department</b> .DepartmentID = HumanResources.EmployeeDepartmentHistory.DepartmentID	Foreign key constraint referencing Department.DepartmentID.

## Unique keys

	Columns	Name / Description
🔑	DepartmentID	PK_Department_DepartmentID Primary key (clustered) constraint
🔑	Name	AK_Department_Name Unique nonclustered index.

### 1.3.1.2. Table: HumanResources.Employee

Employee information such as salary, department, and title.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Primary key for Employee records. Foreign key to BusinessEntity.BusinessEntityID. <b>References:</b> Person.Person
☰	NationalIDNumber	nvarchar(15)	Unique national identification number such as a social security number.
☰	LoginID	nvarchar(256)	Network login. Test2
☰	OrganizationNode	hierarchyid	Where the employee is located in corporate hierarchy. <b>Nullable</b>
☰	OrganizationLevel	smallint	The depth of the employee in the corporate hierarchy. <b>Nullable</b> <b>Computed:</b> [OrganizationNode].[GetLevel]()
☰	JobTitle	nvarchar(50)	Work title such as Buyer or Sales Representative.
☰	BirthDate	date	Date of birth.
☰	MaritalStatus	nchar(1)	M = Married, S = Single
☰	Gender	nchar(1)	M = Male, F = Female
☰	HireDate	date	Employee hired on this date.
☰	SalariedFlag	bit	Job classification. 0 = Hourly, not exempt from collective bargaining. 1 = Salaried, exempt from collective bargaining. <b>Default:</b> 1
☰	VacationHours	smallint	Number of available vacation hours. <b>Default:</b> 0
☰	SickLeaveHours	smallint	Number of available sick leave hours. <b>Default:</b> 0
☰	CurrentFlag	bit	0 = Inactive, 1 = Active <b>Default:</b> 1
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Links to

	Table	Join	Title / Name / Description
→	Person.Person	HumanResources.Employee.BusinessEntityID = Person.Person.BusinessEntityID	FK_Employee_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.

## Linked from

	Table	Join	Title / Name / Description
→	HumanResources.EmployeeDepartmentHistory	<b>HumanResources.Employee.BusinessEntityID</b> = HumanResources.EmployeeDepartmentHistory.BusinessEntityID	FK_EmployeeDepartmentHistory_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.
→	HumanResources.EmployeePayHistory	<b>HumanResources.Employee.BusinessEntityID</b> = HumanResources.EmployeePayHistory.BusinessEntityID	FK_EmployeePayHistory_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.
→	HumanResources.JobCandidate	<b>HumanResources.Employee.BusinessEntityID</b> = HumanResources.JobCandidate.BusinessEntityID	FK_JobCandidate_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.
→	Purchasing.PurchaseOrderHeader	<b>HumanResources.Employee.BusinessEntityID</b> = Purchasing.PurchaseOrderHeader.EmployeeID	FK_PurchaseOrderHeader_Employee_EmployeeID Foreign key constraint referencing Employee.EmployeeID.
→	Sales.SalesPerson	<b>HumanResources.Employee.BusinessEntityID</b> = Sales.SalesPerson.BusinessEntityID	FK_SalesPerson_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.

## Unique keys

	Columns	Name / Description
🔑	BusinessEntityID	PK_Employee_BusinessEntityID Primary key (clustered) constraint
🔑	LoginID	AK_Employee_LoginID Unique nonclustered index.
🔑	NationalIDNumber	AK_Employee_NationalIDNumber Unique nonclustered index.
🔑	rowguid	AK_Employee_rowguid Unique nonclustered index. Used to support replication samples.

## Triggers

	Name	When	Description
⚡	dEmployee	Instead Of Delete	INSTEAD OF DELETE trigger which keeps Employees from being deleted.

### 1.3.1.3. Table: HumanResources.EmployeeDepartmentHistory

Employee department transfers.

## Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Employee identification number. Foreign key to Employee.BusinessEntityID. <b>References:</b> HumanResources.Employee
☰	DepartmentID	smallint	Department in which the employee worked including currently. Foreign key to Department.DepartmentID. <b>References:</b> HumanResources.Department
☰	ShiftID	tinyint	Identifies which 8-hour shift the employee works. Foreign key to Shift.Shift.ID. <b>References:</b> HumanResources.Shift
☰	StartDate	date	Date the employee started work in the department.

	Name	Data type	Description / Attributes
End Date	EndDate	date	Date the employee left the department. NULL = Current department. <b>Nullable</b>
Last update	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	HumanResources.Department	<b>HumanResources.EmployeeDepartmentHistory</b> .DepartmentID = HumanResources.Department.DepartmentID	FK_EmployeeDepartmentHistory_Department_DepartmentID Foreign key constraint referencing Department.DepartmentID.
→	HumanResources.Employee	<b>HumanResources.EmployeeDepartmentHistory</b> .BusinessEntityID = HumanResources.Employee.BusinessEntityID	FK_EmployeeDepartmentHistory_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.
→	HumanResources.Shift	<b>HumanResources.EmployeeDepartmentHistory</b> .ShiftID = HumanResources.Shift.ShiftID	FK_EmployeeDepartmentHistory_Shift_ShiftID Foreign key constraint referencing Shift.ShiftID

Unique keys

	Columns	Name / Description
PK	BusinessEntityID, StartDate, DepartmentID, ShiftID	PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_DepartmentID Primary key (clustered) constraint

#### 1.3.1.4. Table: HumanResources.EmployeePayHistory

Employee pay history.

Columns

	Name	Data type	Description / Attributes
PK	BusinessEntityID	int	Employee identification number. Foreign key to Employee.BusinessEntityID. <b>References:</b> HumanResources.Employee
	RateChangeDate	datetime	Date the change in pay is effective
	Rate	money	Salary hourly rate.
	PayFrequency	tinyint	1 = Salary received monthly, 2 = Salary received biweekly
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	HumanResources.Employee	<b>HumanResources.EmployeePayHistory</b> .BusinessEntityID = HumanResources.Employee.BusinessEntityID	FK_EmployeePayHistory_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.

Unique keys

	Columns	Name / Description
PK	BusinessEntityID, RateChangeDate	PK_EmployeePayHistory_BusinessEntityID_RateChangeDate Primary key (clustered) constraint

### 1.3.1.5. Table: HumanResources.JobCandidate

Résumés submitted to Human Resources by job applicants.

Columns

	Name	Data type	Description / Attributes
█	JobCandidateID	int	Primary key for JobCandidate records. <b>Identity / Auto increment</b>
█	BusinessEntityID	int	Employee identification number if applicant was hired. Foreign key to Employee.BusinessEntityID. <b>Nullable</b> <b>References:</b> HumanResources.Employee
█	Resume	xml	Résumé in XML format. <b>Nullable</b>
█	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	HumanResources.Employee	<b>HumanResources.JobCandidate.BusinessEntityID = HumanResources.Employee.BusinessEntityID</b>	FK_JobCandidate_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.

Unique keys

	Columns	Name / Description
█	JobCandidateID	<b>PK_JobCandidate_JobCandidateID</b> Primary key (clustered) constraint

### 1.3.1.6. Table: HumanResources.Shift

Work shift lookup table.

Columns

	Name	Data type	Description / Attributes
█	ShiftID	tinyint	Primary key for Shift records. <b>Identity / Auto increment</b>
█	Name	nvarchar(50)	Shift description.
█	StartTime	time(7)	Shift start time.
█	EndTime	time(7)	Shift end time.
█	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	HumanResources.EmployeeDepartmentHistory	<b>HumanResources.Shift.ShiftID = HumanResources.EmployeeDepartmentHistory.ShiftID</b>	FK_EmployeeDepartmentHistory_Shift_ShiftID Foreign key constraint referencing Shift.ShiftID

## Unique keys

	Columns	Name / Description
🔑	ShiftID	PK_Shift_ShiftID Primary key (clustered) constraint
🔑	Name	AK_Shift_Name Unique nonclustered index.
🔑	StartTime, EndTime	AK_Shift_StartTime_EndTime Unique nonclustered index.

## 1.3.2. Views

### 1.3.2.1. View: HumanResources.vEmployee

Employee names and addresses.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	
☰	Title	nvarchar(8)	<b>Nullable</b>
☰	FirstName	nvarchar(50)	
☰	MiddleName	nvarchar(50)	<b>Nullable</b>
☰	LastName	nvarchar(50)	
☰	Suffix	nvarchar(10)	<b>Nullable</b>
☰	JobTitle	nvarchar(50)	
☰	PhoneNumber	nvarchar(25)	<b>Nullable</b>
☰	PhoneNumberType	nvarchar(50)	<b>Nullable</b>
☰	EmailAddress	nvarchar(50)	<b>Nullable</b>
☰	EmailPromotion	int	
☰	AddressLine1	nvarchar(60)	
☰	AddressLine2	nvarchar(60)	<b>Nullable</b>
☰	City	nvarchar(30)	
☰	StateProvinceName	nvarchar(50)	
☰	PostalCode	nvarchar(15)	
☰	CountryRegionName	nvarchar(50)	
☰	AdditionalContactInfo	xml	<b>Nullable</b>

### 1.3.2.2. View: HumanResources.vEmployeeDepartment

Returns employee name, title, and current department.

#### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	
☰	Title	nvarchar(8)	<b>Nullable</b>
☰	FirstName	nvarchar(50)	

	Name	Data type	Description / Attributes
■	MiddleName	nvarchar(50)	<b>Nullable</b>
■	LastName	nvarchar(50)	
■	Suffix	nvarchar(10)	<b>Nullable</b>
■	JobTitle	nvarchar(50)	
■	Department	nvarchar(50)	
■	GroupName	nvarchar(50)	
■	StartDate	date	

### 1.3.2.3. View: HumanResources.vEmployeeDepartmentHistory

Returns employee name and current and previous departments.

Columns

	Name	Data type	Description / Attributes
■	BusinessEntityID	int	
■	Title	nvarchar(8)	<b>Nullable</b>
■	FirstName	nvarchar(50)	
■	MiddleName	nvarchar(50)	<b>Nullable</b>
■	LastName	nvarchar(50)	
■	Suffix	nvarchar(10)	<b>Nullable</b>
■	Shift	nvarchar(50)	
■	Department	nvarchar(50)	
■	GroupName	nvarchar(50)	
■	StartDate	date	
■	EndDate	date	<b>Nullable</b>

### 1.3.2.4. View: HumanResources.vJobCandidate

Job candidate names and resumes.

Columns

	Name	Data type	Description / Attributes
■	JobCandidateID	int	<b>Identity / Auto increment</b>
■	BusinessEntityID	int	<b>Nullable</b>
■	Name.Prefix	nvarchar(30)	<b>Nullable</b>
■	Name.First	nvarchar(30)	<b>Nullable</b>
■	Name.Middle	nvarchar(30)	<b>Nullable</b>
■	Name.Last	nvarchar(30)	<b>Nullable</b>
■	Name.Suffix	nvarchar(30)	<b>Nullable</b>
■	Skills	nvarchar(MAX)	<b>Nullable</b>
■	Addr.Type	nvarchar(30)	<b>Nullable</b>
■	Addr.Loc.CountryRegion	nvarchar(100)	<b>Nullable</b>
■	Addr.Loc.State	nvarchar(100)	<b>Nullable</b>

	Name	Data type	Description / Attributes
■	Addr.Loc.City	nvarchar(100)	<b>Nullable</b>
■	Addr.PostalCode	nvarchar(20)	<b>Nullable</b>
■	EMail	nvarchar(MAX)	<b>Nullable</b>
■	WebSite	nvarchar(MAX)	<b>Nullable</b>
■	ModifiedDate	datetime	

### 1.3.2.5. View: HumanResources.vJobCandidateEducation

Displays the content from each education related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.

#### Columns

	Name	Data type	Description / Attributes
■	JobCandidateID	int	<b>Identity / Auto increment</b>
■	Edu.Level	nvarchar(MAX)	<b>Nullable</b>
■	Edu.StartDate	datetime	<b>Nullable</b>
■	Edu.EndDate	datetime	<b>Nullable</b>
■	Edu.Degree	nvarchar(50)	<b>Nullable</b>
■	Edu.Major	nvarchar(50)	<b>Nullable</b>
■	Edu.Minor	nvarchar(50)	<b>Nullable</b>
■	Edu.GPA	nvarchar(5)	<b>Nullable</b>
■	Edu.GPAScale	nvarchar(5)	<b>Nullable</b>
■	Edu.School	nvarchar(100)	<b>Nullable</b>
■	Edu.Loc.CountryRegion	nvarchar(100)	<b>Nullable</b>
■	Edu.Loc.State	nvarchar(100)	<b>Nullable</b>
■	Edu.Loc.City	nvarchar(100)	<b>Nullable</b>

### 1.3.2.6. View: HumanResources.vJobCandidateEmployment

Displays the content from each employment history related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.

#### Columns

	Name	Data type	Description / Attributes
■	JobCandidateID	int	<b>Identity / Auto increment</b>
■	Emp.StartDate	datetime	<b>Nullable</b>
■	Emp.EndDate	datetime	<b>Nullable</b>
■	Emp.OrgName	nvarchar(100)	<b>Nullable</b>
■	Emp.JobTitle	nvarchar(100)	<b>Nullable</b>
■	Emp.Responsibility	nvarchar(MAX)	<b>Nullable</b>
■	Emp.FunctionCategory	nvarchar(MAX)	<b>Nullable</b>
■	Emp.IndustryCategory	nvarchar(MAX)	<b>Nullable</b>

	Name	Data type	Description / Attributes
■	Emp.Loc.CountryRegion	nvarchar(MAX)	Nullable
■	Emp.Loc.State	nvarchar(MAX)	Nullable
■	Emp.Loc.City	nvarchar(MAX)	Nullable

### 1.3.3. Procedures

#### 1.3.3.1. Procedure: dbo.uspGetEmployeeManagers

Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.

##### Input/Output

	Name	Data type	Description
→@	BusinessEntityID	int	Input parameter for the stored procedure uspGetEmployeeManagers. Enter a valid BusinessEntityID from the HumanResources.Employee table.

#### 1.3.3.2. Procedure: dbo.uspGetManagerEmployees

Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.

##### Input/Output

	Name	Data type	Description
→@	BusinessEntityID	int	Input parameter for the stored procedure uspGetManagerEmployees. Enter a valid BusinessEntityID of the manager from the HumanResources.Employee table.

#### 1.3.3.3. Procedure: dbo.uspSearchCandidateResumes

##### Input/Output

	Name	Data type	Description
→@	searchString	nvarchar(1000)	
→@	useInflectional	bit	
→@	useThesaurus	bit	
→@	language	int	

#### 1.3.3.4. Procedure: HumanResources.uspUpdateEmployeeHireInfo

##### Input/Output

	Name	Data type	Description
→@	BusinessEntityID	int	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid BusinessEntityID from the Employee table.
→@	JobTitle	nvarchar(50)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.
→@	HireDate	datetime	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.
→@	RateChangeDate	datetime	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the employee.
→@	Rate	money	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.

	Name	Data type	Description
→@	PayFrequency	tinyint	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.
→@	CurrentFlag	bit	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.

### 1.3.3.5. Procedure: HumanResources.uspUpdateEmployeeLogin

#### Input/Output

	Name	Data type	Description
→@	BusinessEntityID	int	Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.
→@	OrganizationNode	hierarchyid	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.
→@	LoginID	nvarchar(256)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.
→@	JobTitle	nvarchar(50)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.
→@	HireDate	datetime	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.
→@	CurrentFlag	bit	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.

### 1.3.3.6. Procedure: HumanResources.uspUpdateEmployeePersonalInfo

#### Input/Output

	Name	Data type	Description
→@	BusinessEntityID	int	Input parameter for the stored procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the HumanResources.Employee table.
→@	NationalIDNumber	nvarchar(15)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.
→@	BirthDate	datetime	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.
→@	MaritalStatus	nchar(1)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.
→@	Gender	nchar(1)	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.

## 1.3.4. Functions

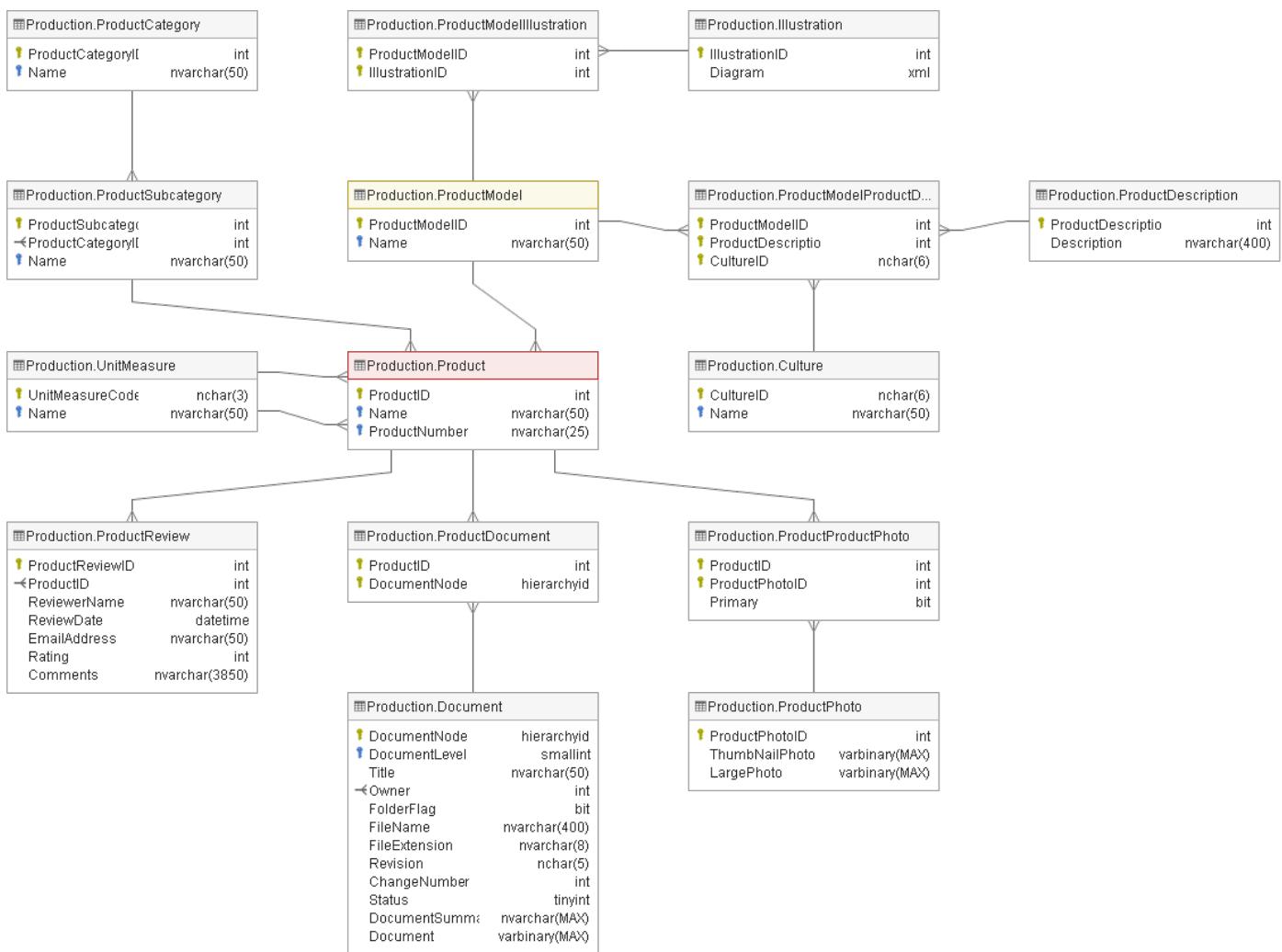
### 1.3.4.1. Function: dbo.ufnGetContactInformation

Table value function returning the first name, last name, job title and contact type for a given contact.

#### Input/Output

	Name	Data type	Description
*@	Returns	table type	
→@	PersonID	int	Input parameter for the table value function ufnGetContactInformation. Enter a valid PersonID from the Person.Contact table.

## 1.4. Products



Products manufactured and sold by Adventure Works Cycles.

### 1.4.1. Tables

#### 1.4.1.1. Table: Production.Culture

Lookup table containing the languages in which some AdventureWorks data is stored.

## Columns

		Name	Data type	Description / Attributes
		CultureID	nchar(6)	Primary key for Culture records.
		Name	nvarchar(50)	Culture description.
		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	Production.ProductModelProductDescriptionCulture	<b>Production.Culture.CultureID =</b> Production.ProductModelProductDescriptionCulture.CultureID	FK_ProductModelProductDescriptionCulture_Culture_CultureID Foreign key constraint referencing Culture.CultureID.

## Unique keys

	Columns	Name / Description
🔑	CultureID	PK_Culture_CultureID Primary key (clustered) constraint
🔑	Name	AK_Culture_Name Unique nonclustered index.

## 1.4.1.2. Table: Production.Document

Product maintenance documents.

### Columns

	Name	Data type	Description / Attributes
☰	DocumentNode	hierarchyid	Primary key for Document records.
☰	DocumentLevel	smallint	Depth in the document hierarchy. <b>Nullable</b> <b>Computed:</b> [DocumentNode].[GetLevel]()
☰	Title	nvarchar(50)	Title of the document.
☰	Owner	int	Employee who controls the document. Foreign key to Employee.BusinessEntityID
☰	FolderFlag	bit	0 = This is a folder, 1 = This is a document. <b>Default:</b> 0
☰	FileName	nvarchar(400)	File name of the document
☰	FileExtension	nvarchar(8)	File extension indicating the document type. For example, .doc or .txt.
☰	Revision	nchar(5)	Revision number of the document.
☰	ChangeNumber	int	Engineering change approval number. <b>Default:</b> 0
☰	Status	tinyint	1 = Pending approval, 2 = Approved, 3 = Obsolete
☰	DocumentSummary	nvarchar(MAX)	Document abstract. <b>Nullable</b>
☰	Document	varbinary(MAX)	Complete document. <b>Nullable</b>
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Required for FileStream. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

### Linked from

	Table	Join	Title / Name / Description
→	Production.ProductDocument	<b>Production.Document</b> .DocumentNode = Production.ProductDocument.DocumentNode	FK_ProductDocument_Document_DocumentNode Foreign key constraint referencing Document.DocumentNode.

### Unique keys

	Columns	Name / Description
🔑	DocumentNode	PK_Document_DocumentNode Primary key (clustered) constraint

	Columns	Name / Description
key	DocumentLevel, DocumentNode	AK_Document_DocumentLevel_DocumentNode Unique nonclustered index.
key	rowguid	AK_Document_rowguid Unique nonclustered index. Used to support FileStream.
key	rowguid	UQ_Document_F73921F793071A63

#### 1.4.1.3. Table: Production.Illustration

Bicycle assembly diagrams.

Columns

	Name	Data type	Description / Attributes
key	IllustrationID	int	Primary key for Illustration records. <b>Identity / Auto increment</b>
key	Diagram	xml	Illustrations used in manufacturing instructions. Stored as XML. <b>Nullable</b>
key	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	Production.ProductModelIllustration	<b>Production.Illustration.IllustrationID =</b> Production.ProductModelIllustration.IllustrationID	FK_ProductModelIllustration_Illustration_IllustrationID Foreign key constraint referencing Illustration.IllustrationID.

Unique keys

	Columns	Name / Description
key	IllustrationID	PK_Illustration_IllustrationID Primary key (clustered) constraint

#### 1.4.1.4. Table: Production.Product

Products sold or used in the manufacturing of sold products.

Columns

	Name	Data type	Description / Attributes
key	ProductID	int	Primary key for Product records. <b>Identity / Auto increment</b>
key	Name	nvarchar(50)	Name of the product.
key	ProductNumber	nvarchar(25)	Unique product identification number.
key	MakeFlag	bit	0 = Product is purchased, 1 = Product is manufactured in-house. <b>Default:</b> 1
key	FinishedGoodsFlag	bit	0 = Product is not a salable item. 1 = Product is salable. <b>Default:</b> 1
key	Color	nvarchar(15)	Product color. <b>Nullable</b>
key	SafetyStockLevel	smallint	Minimum inventory quantity.
key	ReorderPoint	smallint	Inventory level that triggers a purchase order or work order.
key	StandardCost	money	Standard cost of the product.

	Name	Data type	Description / Attributes
>ListPrice	money		Selling price.
Size	nvarchar(5)		Product size. <b>Nullable</b>
SizeUnitMeasureCode	nchar(3)		Unit of measure for Size column. <b>Nullable</b> <b>References:</b> Production.UnitMeasure
WeightUnitMeasureCode	nchar(3)		Unit of measure for Weight column. <b>Nullable</b> <b>References:</b> Production.UnitMeasure
Weight	decimal(8, 2)		Product weight. <b>Nullable</b>
DaysToManufacture	int		Number of days required to manufacture the product.
ProductLine	nchar(2)		R = Road, M = Mountain, T = Touring, S = Standard <b>Nullable</b>
Class	nchar(2)		H = High, M = Medium, L = Low <b>Nullable</b>
Style	nchar(2)		W = Womens, M = Mens, U = Universal <b>Nullable</b>
ProductSubcategoryID	int		Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID. <b>Nullable</b> <b>References:</b> Production.ProductSubcategory
ProductModelID	int		Product is a member of this product model. Foreign key to ProductModel.ProductModelID. <b>Nullable</b> <b>References:</b> Production.ProductModel
SellStartDate	datetime		Date the product was available for sale.
SellEndDate	datetime		Date the product was no longer available for sale. <b>Nullable</b>
DiscontinuedDate	datetime		Date the product was discontinued. <b>Nullable</b>
rowguid	uniqueidentifier		ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
ModifiedDate	datetime		Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.ProductModel	<b>Production.Product.ProductModelID</b> = Production.ProductModel.ProductModelID	FK_Product_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.
→	Production.ProductSubcategory	<b>Production.Product.ProductSubcategoryID</b> = Production.ProductSubcategory.ProductSubcategoryID	FK_Product_ProductSubcategory_ProductSubcategoryID Foreign key constraint referencing ProductSubcategory.ProductSubcategoryID.
→	Production.UnitMeasure	<b>Production.Product.SizeUnitMeasureCode</b> = Production.UnitMeasure.UnitMeasureCode	FK_Product_UnitMeasure_SizeUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.

	Table	Join	Title / Name / Description
→	Production.UnitMeasure	<b>Production.Product.WeightUnitMeasureCode</b> = Production.UnitMeasure.UnitMeasureCode	FK_Product_UnitMeasure_WeightUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.

### Linked from

	Table	Join	Title / Name / Description
←	Production.BillOfMaterials	<b>Production.Product.ProductID</b> = Production.BillOfMaterials.ComponentID	FK_BillOfMaterials_Product_ComponentID Foreign key constraint referencing Product.ProductAssemblyID.
←	Production.BillOfMaterials	<b>Production.Product.ProductID</b> = Production.BillOfMaterials.ProductAssemblyID	FK_BillOfMaterials_Product_ProductAssemblyID Foreign key constraint referencing Product.ProductAssemblyID.
←	Production.ProductCostHistory	<b>Production.Product.ProductID</b> = Production.ProductCostHistory.ProductID	FK_ProductCostHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Production.ProductDocument	<b>Production.Product.ProductID</b> = Production.ProductDocument.ProductID	FK_ProductDocument_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Production.ProductInventory	<b>Production.Product.ProductID</b> = Production.ProductInventory.ProductID	FK_ProductInventory_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Production.ProductListPriceHistory	<b>Production.Product.ProductID</b> = Production.ProductListPriceHistory.ProductID	FK_ProductListPriceHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Production.ProductProductPhoto	<b>Production.Product.ProductID</b> = Production.ProductProductPhoto.ProductID	FK_ProductProductPhoto_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Production.ProductReview	<b>Production.Product.ProductID</b> = Production.ProductReview.ProductID	FK_ProductReview_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Purchasing.ProductVendor	<b>Production.Product.ProductID</b> = Purchasing.ProductVendor.ProductID	FK_ProductVendor_Product_ProductID Foreign key constraint referencing Product.ProductID.
←	Purchasing.PurchaseOrderDetail	<b>Production.Product.ProductID</b> = Purchasing.PurchaseOrderDetail.ProductID	FK_PurchaseOrderDetail_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Sales.SalesOrderDetail	<b>Production.Product.ProductID</b> = Sales.SalesOrderDetail.ProductID	User-defined relation
→	Sales.ShoppingCartItem	<b>Production.Product.ProductID</b> = Sales.ShoppingCartItem.ProductID	FK_ShoppingCartItem_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Sales.SpecialOfferProduct	<b>Production.Product.ProductID</b> = Sales.SpecialOfferProduct.ProductID	FK_SpecialOfferProduct_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Production.TransactionHistory	<b>Production.Product.ProductID</b> = Production.TransactionHistory.ProductID	FK_TransactionHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Production.TransactionHistoryArchive	<b>Production.Product.ProductID</b> = Production.TransactionHistoryArchive.ProductID	User-defined relation
→	Production.WorkOrder	<b>Production.Product.ProductID</b> = Production.WorkOrder.ProductID	FK_WorkOrder_Product_ProductID Foreign key constraint referencing Product.ProductID.

## Unique keys

	Columns	Name / Description
🔑	ProductID	PK_Product_ProductID Primary key (clustered) constraint
🔑	Name	AK_Product_Name Unique nonclustered index.
🔑	ProductNumber	AK_Product_ProductNumber Unique nonclustered index.
🔑	rowguid	AK_Product_rowguid Unique nonclustered index. Used to support replication samples.

## 1.4.1.5. Table: Production.ProductCategory

High-level product categorization.

### Columns

	Name	Data type	Description / Attributes
☰	ProductCategoryID	int	Primary key for ProductCategory records. <b>Identity / Auto increment</b>
☰	Name	nvarchar(50)	Category description.
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

### Linked from

	Table	Join	Title / Name / Description
→	Production.ProductSubcategory	Production.ProductCategory.ProductCategoryID = Production.ProductSubcategory.ProductCategoryID	FK_ProductSubcategory_ProductCategory_ProductCategoryID Foreign key constraint referencing ProductCategory.ProductCategoryID.

## Unique keys

	Columns	Name / Description
🔑	ProductCategoryID	PK_ProductCategory_ProductCategoryID Primary key (clustered) constraint
🔑	Name	AK_ProductCategory_Name Unique nonclustered index.
🔑	rowguid	AK_ProductCategory_rowguid Unique nonclustered index. Used to support replication samples.

## 1.4.1.6. Table: Production.ProductDescription

Product descriptions in several languages.

### Columns

	Name	Data type	Description / Attributes
☰	ProductDescriptionID	int	Primary key for ProductDescription records. <b>Identity / Auto increment</b>
☰	Description	nvarchar(400)	Description of the product.

		Name	Data type	Description / Attributes
█	🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	Production.ProductModelProductDescriptionCulture	<b>Production.ProductDescription.ProductDescriptionID = Production.ProductModelProductDescriptionCulture.ProductDescriptionID</b>	FK_ProductModelProductDescriptionCulture_ProductDescription_Prod uctDescriptionID Foreign key constraint referencing ProductDescription.ProductDescriptionID.

Unique keys

	Columns	Name / Description
🔑	ProductDescriptionID	PK_ProductDescription_ProductDescriptionID Primary key (clustered) constraint
🔑	rowguid	AK_ProductDescription_rowguid Unique nonclustered index. Used to support replication samples.

#### 1.4.1.7. Table: Production.ProductDocument

Cross-reference table mapping products to related product documents.

Columns

		Name	Data type	Description / Attributes
█	🔑	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
█	🔑	DocumentNode	hierarchyid	Document identification number. Foreign key to Document.DocumentNode. <b>References:</b> Production.Document
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Document	<b>Production.ProductDocument.DocumentNode = Production.Document.DocumentNode</b>	FK_ProductDocument_Document_DocumentNode Foreign key constraint referencing Document.DocumentNode.
→	Production.Product	<b>Production.ProductDocument.ProductID = Production.Product.ProductID</b>	FK_ProductDocument_Product_ProductID Foreign key constraint referencing Product.ProductID.

Unique keys

	Columns	Name / Description
🔑	ProductID, DocumentNode	PK_ProductDocument_ProductID_DocumentNode Primary key (clustered) constraint

#### 1.4.1.8. Table: Production.ProductModel

Product model classification.

## Columns

		Name	Data type	Description / Attributes
■	🔑	ProductModelID	int	Primary key for ProductModel records. <b>Identity / Auto increment</b>
■	🔑	Name	nvarchar(50)	Product model description.
■		CatalogDescription	xml	Detailed product catalog information in xml format. <b>Nullable</b>
■		Instructions	xml	Manufacturing instructions in xml format. <b>Nullable</b>
■	🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
■		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.ProductModel</b> .ProductModelID = Production.Product.ProductModelID	FK_Product_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.
→	Production.ProductModelIllustration	<b>Production.ProductModel</b> .ProductModelID = Production.ProductModelIllustration.ProductModelID	FK_ProductModelIllustration_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.
→	Production.ProductModelProductDescriptionCulture	<b>Production.ProductModel</b> .ProductModelID = Production.ProductModelProductDescriptionCulture.ProductModelID	FK_ProductModelProductDescriptionCulture_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.

## Unique keys

	Columns	Name / Description
🔑	ProductModelID	PK_ProductModel_ProductModelID Primary key (clustered) constraint
🔑	Name	AK_ProductModel_Name Unique nonclustered index.
🔑	rowguid	AK_ProductModel_rowguid Unique nonclustered index. Used to support replication samples.

## 1.4.1.9. Table: Production.ProductModelIllustration

Cross-reference table mapping product models and illustrations.

## Columns

		Name	Data type	Description / Attributes
■	🔑	ProductModelID	int	Primary key. Foreign key to ProductModel.ProductModelID. <b>References:</b> Production.ProductModel
■	🔑	IllustrationID	int	Primary key. Foreign key to Illustration.IllustrationID. <b>References:</b> Production.Illustration
■		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Production.Illustration	<b>Production.ProductModelIllustration.IllustrationID =</b> Production.Illustration.IllustrationID	FK_ProductModelIllustration_Illustration_IllustrationID Foreign key constraint referencing Illustration.IllustrationID.
→	Production.ProductModel	<b>Production.ProductModelIllustration.ProductModelID =</b> Production.ProductModel.ProductModelID	FK_ProductModelIllustration_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.

## Unique keys

	Columns	Name / Description
🔑	ProductModelID, IllustrationID	PK_ProductModelIllustration_ProductModelID_IllustrationID Primary key (clustered) constraint

## 1.4.1.10. Table: Production.ProductModelProductDescriptionCulture

Cross-reference table mapping product descriptions and the language the description is written in.

### Columns

	Name	Data type	Description / Attributes
.PrimaryKey	ProductModelID	int	Primary key. Foreign key to ProductModel.ProductModelID. <b>References:</b> Production.ProductModel
PrimaryKey	ProductDescriptionID	int	Primary key. Foreign key to ProductDescription.ProductDescriptionID. <b>References:</b> Production.ProductDescription
PrimaryKey	CultureID	nchar(6)	Culture identification number. Foreign key to Culture.CultureID. <b>References:</b> Production.Culture
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Production.Culture	<b>Production.ProductModelProductDescriptionCulture.CultureID =</b> Production.Culture.CultureID	FK_ProductModelProductDescriptionCulture_Culture_CultureID Foreign key constraint referencing Culture.CultureID.
→	Production.ProductDescription	<b>Production.ProductModelProductDescriptionCulture.ProductDescriptionID =</b> Production.ProductDescription.ProductDescriptionID	FK_ProductModelProductDescriptionCulture_ProductDescription_Prod uctDescriptionID Foreign key constraint referencing ProductDescription.ProductDescriptionID.
→	Production.ProductModel	<b>Production.ProductModelProductDescriptionCulture.ProductModelID =</b> Production.ProductModel.ProductModelID	FK_ProductModelProductDescriptionCulture_ProductModel_ProductM odelID Foreign key constraint referencing ProductModel.ProductModelID.

## Unique keys

	Columns	Name / Description
🔑	ProductModelID, ProductDescriptionID, CultureID	PK_ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID Primary key (clustered) constraint

## 1.4.1.11. Table: Production.ProductPhoto

Product images.

## Columns

	Name	Data type	Description / Attributes
PK	ProductPhotoID	int	Primary key for ProductPhoto records. <b>Identity / Auto increment</b>
IM	ThumbNailPhoto	varbinary(MAX)	Small image of the product. <b>Nullable</b>
IM	ThumbnailPhotoFileName	nvarchar(50)	Small image file name. <b>Nullable</b>
IM	LargePhoto	varbinary(MAX)	Large image of the product. <b>Nullable</b>
IM	LargePhotoFileName	nvarchar(50)	Large image file name. <b>Nullable</b>
IM	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Production.ProductProductPhoto	<b>Production.ProductPhoto</b> .ProductPhotoID = Production.ProductProductPhoto.ProductPhotoID	FK_ProductProductPhoto_ProductPhoto_ProductPhotoID Foreign key constraint referencing ProductPhoto.ProductPhotoID.

## Unique keys

	Columns	Name / Description
PK	ProductPhotoID	PK_ProductPhoto_ProductPhotoID Primary key (clustered) constraint

## 1.4.1.12. Table: Production.ProductProductPhoto

Cross-reference table mapping products and product photos.

## Columns

	Name	Data type	Description / Attributes
PK	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
PK	ProductPhotoID	int	Product photo identification number. Foreign key to ProductPhoto.ProductPhotoID. <b>References:</b> Production.ProductPhoto
IM	Primary	bit	0 = Photo is not the principal image. 1 = Photo is the principal image. <b>Default:</b> 0
IM	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.ProductProductPhoto</b> .ProductID = Production.Product.ProductID	FK_ProductProductPhoto_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Production.ProductPhoto	<b>Production.ProductProductPhoto</b> .ProductPhotoID = Production.ProductPhoto.ProductPhotoID	FK_ProductProductPhoto_ProductPhoto_ProductPhotoID Foreign key constraint referencing ProductPhoto.ProductPhotoID.

## Unique keys

	Columns	Name / Description
🔑	ProductID, ProductPhotoID	PK_ProductProductPhoto_ProductID_ProductPhotoID Primary key (clustered) constraint

### 1.4.1.13. Table: Production.ProductReview

Customer reviews of products they have purchased.

#### Columns

	Name	Data type	Description / Attributes
☰	ProductReviewID	int	Primary key for ProductReview records. <b>Identity / Auto increment</b>
☰	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
☰	ReviewerName	nvarchar(50)	Name of the reviewer.
☰	ReviewDate	datetime	Date review was submitted. <b>Default:</b> getdate()
☰	EmailAddress	nvarchar(50)	Reviewer's e-mail address.
☰	Rating	int	Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.
☰	Comments	nvarchar(3850)	Reviewer's comments <b>Nullable</b>
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Links to

	Table	Join	Title / Name / Description
→	Production.Product	Production.ProductReview.ProductID = Production.Product.ProductID	FK_ProductReview_Product_ProductID Foreign key constraint referencing Product.ProductID.

#### Unique keys

	Columns	Name / Description
🔑	ProductReviewID	PK_ProductReview_ProductReviewID Primary key (clustered) constraint

### 1.4.1.14. Table: Production.ProductSubcategory

Product subcategories. See ProductCategory table.

#### Columns

	Name	Data type	Description / Attributes
☰	ProductSubcategoryID	int	Primary key for ProductSubcategory records. <b>Identity / Auto increment</b>
☰	ProductCategoryID	int	Product category identification number. Foreign key to ProductCategory.ProductCategoryID. <b>References:</b> Production.ProductCategory
☰	Name	nvarchar(50)	Subcategory description.
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()

		Name	Data type	Description / Attributes
█	ModifiedDate		datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.ProductCategory	<b>Production.ProductSubcategory</b> .ProductCategoryID = Production.ProductCategory.ProductCategoryID	FK_ProductSubcategory_ProductCategory_ProductCategoryID Foreign key constraint referencing ProductCategory.ProductCategoryID.

Linked from

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.ProductSubcategory</b> .ProductSubcategoryId = Production.Product.ProductSubcategoryId	FK_Product_ProductSubcategory_ProductSubcategoryId Foreign key constraint referencing ProductSubcategory.ProductSubcategoryId.

Unique keys

	Columns	Name / Description
🔑	ProductSubcategoryId	PK_ProductSubcategory_ProductSubcategoryId Primary key (clustered) constraint
🔑	Name	AK_ProductSubcategory_Name Unique nonclustered index.
🔑	rowguid	AK_ProductSubcategory_rowguid Unique nonclustered index. Used to support replication samples.

#### 1.4.1.15. Table: Production.UnitMeasure

Unit of measure lookup table.

Columns

		Name	Data type	Description / Attributes
█	🔑	UnitMeasureCode	nchar(3)	Primary key.
█	🔑	Name	nvarchar(50)	Unit of measure description.
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	Production.BillOfMaterials	<b>Production.UnitMeasure</b> .UnitMeasureCode = Production.BillOfMaterials.UnitMeasureCode	FK_BillOfMaterials_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
→	Production.Product	<b>Production.UnitMeasure</b> .UnitMeasureCode = Production.Product.SizeUnitMeasureCode	FK_Product_UnitMeasure_SizeUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
→	Production.Product	<b>Production.UnitMeasure</b> .UnitMeasureCode = Production.Product.WeightUnitMeasureCode	FK_Product_UnitMeasure_WeightUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.

	Table	Join	Title / Name / Description
→	Purchasing.ProductVendor	Production.UnitMeasure.UnitMeasure Code = Purchasing.ProductVendor.UnitMeasu reCode	FK_ProductVendor_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.

## Unique keys

	Columns	Name / Description
🔑	UnitMeasureCode	PK_UnitMeasure_UnitMeasureCode Primary key (clustered) constraint
🔑	Name	AK_UnitMeasure_Name Unique nonclustered index.

## 1.4.2. Views

### 1.4.2.1. View: Production.vProductAndDescription

Product names and descriptions. Product descriptions are provided in multiple languages.

#### Columns

	Name	Data type	Description / Attributes
☰	ProductID	int	
☰	Name	nvarchar(50)	
☰	ProductModel	nvarchar(50)	
☰	CultureID	nchar(6)	
☰	Description	nvarchar(400)	

### 1.4.2.2. View: Production.vProductModelCatalogDescription

Displays the content from each element in the xml column CatalogDescription for each product in the Production.ProductModel table that has catalog data.

#### Columns

	Name	Data type	Description / Attributes
☰	ProductModelID	int	Identity / Auto increment
☰	Name	nvarchar(50)	
☰	Summary	nvarchar(MAX)	Nullable
☰	Manufacturer	nvarchar(MAX)	Nullable
☰	Copyright	nvarchar(30)	Nullable
☰	ProductURL	nvarchar(256)	Nullable
☰	WarrantyPeriod	nvarchar(256)	Nullable
☰	WarrantyDescription	nvarchar(256)	Nullable
☰	NoOfYears	nvarchar(256)	Nullable
☰	MaintenanceDescription	nvarchar(256)	Nullable
☰	Wheel	nvarchar(256)	Nullable
☰	Saddle	nvarchar(256)	Nullable
☰	Pedal	nvarchar(256)	Nullable

	Name	Data type	Description / Attributes
■	BikeFrame	nvarchar(MAX)	<b>Nullable</b>
■	Crankset	nvarchar(256)	<b>Nullable</b>
■	PictureAngle	nvarchar(256)	<b>Nullable</b>
■	PictureSize	nvarchar(256)	<b>Nullable</b>
■	ProductPhotoID	nvarchar(256)	<b>Nullable</b>
■	Material	nvarchar(256)	<b>Nullable</b>
■	Color	nvarchar(256)	<b>Nullable</b>
■	ProductLine	nvarchar(256)	<b>Nullable</b>
■	Style	nvarchar(256)	<b>Nullable</b>
■	RiderExperience	nvarchar(1024)	<b>Nullable</b>
■	rowguid	uniqueidentifier	
■	ModifiedDate	datetime	

#### 1.4.2.3. View: Production.vProductModelInstructions

Displays the content from each element in the xml column Instructions for each product in the Production.ProductModel table that has manufacturing instructions.

#### Columns

	Name	Data type	Description / Attributes
■	ProductModelID	int	<b>Identity / Auto increment</b>
■	Name	nvarchar(50)	
■	Instructions	nvarchar(MAX)	<b>Nullable</b>
■	LocationID	int	<b>Nullable</b>
■	SetupHours	decimal(9, 4)	<b>Nullable</b>
■	MachineHours	decimal(9, 4)	<b>Nullable</b>
■	LaborHours	decimal(9, 4)	<b>Nullable</b>
■	LotSize	int	<b>Nullable</b>
■	Step	nvarchar(1024)	<b>Nullable</b>
■	rowguid	uniqueidentifier	
■	ModifiedDate	datetime	

#### 1.4.3. Procedures

##### 1.4.3.1. Procedure: dbo.uspGetBillOfMaterials

Uses a recursive query (common table expression) to generate a multilevel Bill of Material: all level 1 components of a level 0 assembly, all level 2 components of a level 1 assembly, and so on.

#### Input/Output

	Name	Data type	Description
→@	StartProductID	int	Input parameter for the stored procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product table.
→@	CheckDate	datetime	Input parameter for the stored procedure uspGetBillOfMaterials used to eliminate components not used after that date. Enter a valid date.

#### 1.4.3.2. Procedure: dbo.uspGetWhereUsedProductID

Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified ProductID.

##### Input/Output

	Name	Data type	Description
→@	StartProductID	int	Input parameter for the stored procedure uspGetWhereUsedProductID. Enter a valid ProductID from the Production.Product table.
→@	CheckDate	datetime	Input parameter for the stored procedure uspGetWhereUsedProductID used to eliminate components not used after that date. Enter a valid date.

#### 1.4.4. Functions

##### 1.4.4.1. Function: dbo.ufnGetDocumentStatusText

Scalar function returning the text representation of the Status column in the Document table.

##### Input/Output

	Name	Data type	Description
*@	Returns	nvarchar(16)	Returns the text representation of the Status column
→@	Status	tinyint	Input parameter for the scalar function ufnGetDocumentStatusText. Enter a valid integer.

##### 1.4.4.2. Function: dbo.ufnGetProductDealerPrice

Scalar function returning the dealer price for a given product on a particular order date.

##### Input/Output

	Name	Data type	Description
*@	Returns	money	
→@	ProductID	int	Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid ProductID from the Production.Product table.
→@	OrderDate	datetime	Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid order date.

##### 1.4.4.3. Function: dbo.ufnGetProductListPrice

Scalar function returning the list price for a given product on a particular order date.

##### Input/Output

	Name	Data type	Description
*@	Returns	money	
→@	ProductID	int	Input parameter for the scalar function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product table.
→@	OrderDate	datetime	Input parameter for the scalar function ufnGetProductListPrice. Enter a valid order date.

##### 1.4.4.4. Function: dbo.ufnGetProductStandardCost

Scalar function returning the standard cost for a given product on a particular order date.

## Input/Output

	Name	Data type	Description
*@	Returns	money	
*@	ProductID	int	Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid ProductID from the Production.Product table.
*@	OrderDate	datetime	Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid order date.

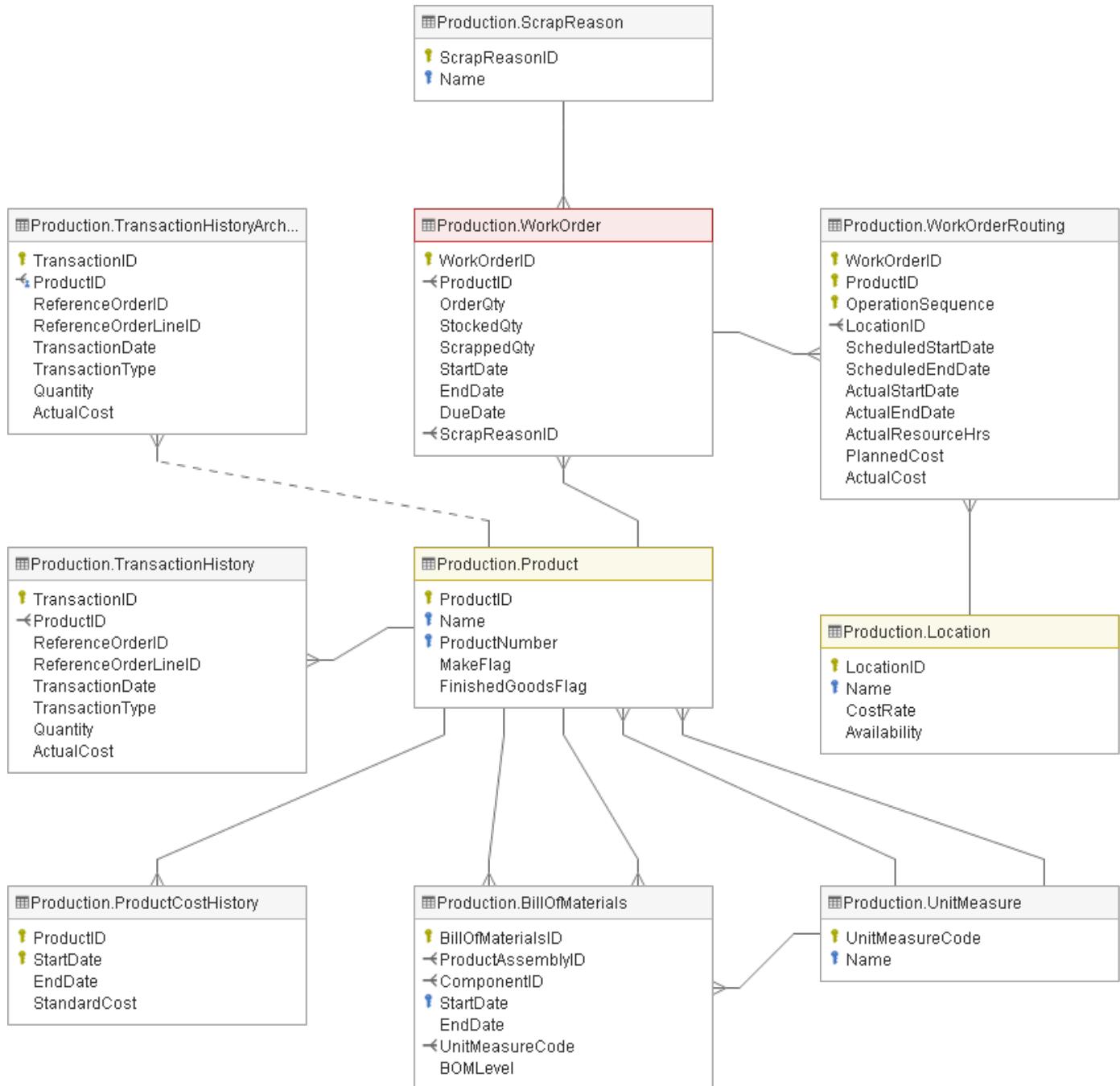
## 1.4.4.5. Function: dbo.ufnGetStock

Scalar function returning the quantity of inventory in LocationID 6 (Miscellaneous Storage)for a specified ProductID.

## Input/Output

	Name	Data type	Description
*@	Returns	int	
*@	ProductID	int	Input parameter for the scalar function ufnGetStock. Enter a valid ProductID from the Production.ProductInventory table.

## 1.5. Manufacturing



Manufacturing process management module.

### 1.5.1. Tables

#### 1.5.1.1. Table: Production.BillOfMaterials

Items required to make bicycles and bicycle subassemblies. It identifies the hierarchical relationship between a parent product and its components.

##### Columns

	Name	Data type	Description / Attributes
	BillOfMaterialsID	int	Primary key for BillOfMaterials records. Identity / Auto increment

		Name	Data type	Description / Attributes
■	🔑	ProductAssemblyID	int	Parent product identification number. Foreign key to Product.ProductID. <b>Nullable</b> <b>References:</b> Production.Product
■	🔑	ComponentID	int	Component identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
■	🔑	StartDate	datetime	Date the component started being used in the assembly item. <b>Default:</b> getdate()
■		EndDate	datetime	Date the component stopped being used in the assembly item. <b>Nullable</b>
■		UnitMeasureCode	nchar(3)	Standard code identifying the unit of measure for the quantity. <b>References:</b> Production.UnitMeasure
■		BOMLevel	smallint	Indicates the depth the component is from its parent (AssemblyID).
■		PerAssemblyQty	decimal(8, 2)	Quantity of the component needed to create the assembly. <b>Default:</b> 1.00
■		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.BillOfMaterials.Component</b> ID = Production.Product.ProductID	FK_BillOfMaterials_Product_ComponentID Foreign key constraint referencing Product.ProductAssemblyID.
→	Production.Product	<b>Production.BillOfMaterials.ProductAssemblyID</b> = Production.Product.ProductID	FK_BillOfMaterials_Product_ProductAssemblyID Foreign key constraint referencing Product.ProductAssemblyID.
→	Production.UnitMeasure	<b>Production.BillOfMaterials.UnitMeasureCode</b> = Production.UnitMeasure.UnitMeasureCode	FK_BillOfMaterials_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.

Unique keys

	Columns	Name / Description
🔑	BillOfMaterialsID	PK_BillOfMaterials_BillOfMaterialsID Primary key (clustered) constraint
🔑	ProductAssemblyID, ComponentID, StartDate	AK_BillOfMaterials_ProductAssemblyID_ComponentID_StartDate Clustered index.

### 1.5.1.2. Table: Production.ProductCostHistory

Changes in the cost of a product over time.

Columns

		Name	Data type	Description / Attributes
■	🔑	ProductID	int	Product identification number. Foreign key to Product.ProductID <b>References:</b> Production.Product
■	🔑	StartDate	datetime	Product cost start date.
■		EndDate	datetime	Product cost end date. <b>Nullable</b>
■		StandardCost	money	Standard cost of the product.
■		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.ProductCostHistory</b> .ProductID = Production.Product.ProductID	FK_ProductCostHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.

Unique keys

	Columns	Name / Description
🔑	ProductID, StartDate	PK_ProductCostHistory_ProductID_StartDate Primary key (clustered) constraint

### 1.5.1.3. Table: Production.ProductListPriceHistory

Changes in the list price of a product over time.

Columns

	Name	Data type	Description / Attributes
☰	ProductID	int	Product identification number. Foreign key to Product.ProductID <b>References:</b> Production.Product
☰	StartDate	datetime	List price start date.
☰	EndDate	datetime	List price end date <b>Nullable</b>
☰	ListPrice	money	Product list price.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.ProductListPriceHistory</b> .ProductID = Production.Product.ProductID	FK_ProductListPriceHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.

Unique keys

	Columns	Name / Description
🔑	ProductID, StartDate	PK_ProductListPriceHistory_ProductID_StartDate Primary key (clustered) constraint

### 1.5.1.4. Table: Production.ScrapReason

Manufacturing failure reasons lookup table.

Columns

	Name	Data type	Description / Attributes
☰	ScrapReasonID	smallint	Primary key for ScrapReason records. <b>Identity / Auto increment</b>
☰	Name	nvarchar(50)	Failure description.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Production.WorkOrder	<b>Production.ScrapReason.ScrapReasonID</b> D = Production.WorkOrder.ScrapReasonID	FK_WorkOrder_ScrapReason_ScrapReasonID Foreign key constraint referencing ScrapReason.ScrapReasonID.

## Unique keys

	Columns	Name / Description
🔑	ScrapReasonID	PK_ScrapReason_ScrapReasonID Primary key (clustered) constraint
🔑	Name	AK_ScrapReason_Name Unique nonclustered index.

## 1.5.1.5. Table: Production.TransactionHistory

Record of each purchase order, sales order, or work order transaction year to date.

### Columns

	Name	Data type	Description / Attributes
>ID	TransactionID	int	Primary key for TransactionHistory records. <b>Identity / Auto increment</b>
ID	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
ID	ReferenceOrderID	int	Purchase order, sales order, or work order identification number.
ID	ReferenceOrderLineID	int	Line number associated with the purchase order, sales order, or work order. <b>Default:</b> 0
ID	TransactionDate	datetime	Date and time of the transaction. <b>Default:</b> getdate()
ID	TransactionType	nchar(1)	W = WorkOrder, S = SalesOrder, P = PurchaseOrder
ID	Quantity	int	Product quantity.
ID	ActualCost	money	Product cost.
ID	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.TransactionHistory.Product</b> ID = Production.Product.ProductID	FK_TransactionHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.

## Unique keys

	Columns	Name / Description
🔑	TransactionID	PK_TransactionHistory_TransactionID Primary key (clustered) constraint

## 1.5.1.6. Table: Production.TransactionHistoryArchive

Transactions for previous years.

## Columns

	Name	Data type	Description / Attributes
PK	TransactionID	int	Primary key for TransactionHistoryArchive records.
FK	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
FK	ReferenceOrderID	int	Purchase order, sales order, or work order identification number.
FK	ReferenceOrderLineID	int	Line number associated with the purchase order, sales order, or work order. <b>Default:</b> 0
FK	TransactionDate	datetime	Date and time of the transaction. <b>Default:</b> getdate()
FK	TransactionType	nchar(1)	W = Work Order, S = Sales Order, P = Purchase Order
FK	Quantity	int	Product quantity.
FK	ActualCost	money	Product cost.
FK	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
FK	Production.Product	<b>Production.TransactionHistoryArchive.</b> ProductID = Production.Product.ProductID	User-defined relation

## Unique keys

	Columns	Name / Description
PK	TransactionID	PK_TransactionHistoryArchive_TransactionID Primary key (clustered) constraint

## 1.5.1.7. Table: Production.WorkOrder

Manufacturing work orders.

## Columns

	Name	Data type	Description / Attributes
PK	WorkOrderID	int	Primary key for WorkOrder records. <b>Identity / Auto increment</b>
FK	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
FK	OrderQty	int	Product quantity to build.
FK	StockedQty	int	Quantity built and put in inventory. <b>Computed:</b> isnull([OrderQty]-[ScrappedQty],(0))
FK	ScrappedQty	smallint	Quantity that failed inspection.
FK	StartDate	datetime	Work order start date.
FK	EndDate	datetime	Work order end date. <b>Nullable</b>
FK	DueDate	datetime	Work order due date.
FK	ScrapReasonID	smallint	Reason for inspection failure. <b>Nullable</b> <b>References:</b> Production.ScrapReason

		Name	Data type	Description / Attributes
🕒	ModifiedDate		datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Production.WorkOrder</b> .ProductID = Production.Product.ProductID	FK_WorkOrder_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Production.ScrapReason	<b>Production.WorkOrder</b> .ScrapReasonID = Production.ScrapReason.ScrapReasonID	FK_WorkOrder_ScrapReason_ScrapReasonID Foreign key constraint referencing ScrapReason.ScrapReasonID.

Linked from

	Table	Join	Title / Name / Description
←	Production.WorkOrderRouting	<b>Production.WorkOrder</b> .WorkOrderID = Production.WorkOrderRouting.WorkOrderID	FK_WorkOrderRouting_WorkOrder_WorkOrderID Foreign key constraint referencing WorkOrder.WorkOrderID.

Unique keys

	Columns	Name / Description
🔑	WorkOrderID	PK_WorkOrder_WorkOrderID Primary key (clustered) constraint

Triggers

	Name	When	Description
⚡	iWorkOrder	After Insert	AFTER INSERT trigger that inserts a row in the TransactionHistory table.
⚡	uWorkOrder	After Update	AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in the WorkOrder table.

### 1.5.1.8. Table: Production.WorkOrderRouting

Work order details.

Columns

		Name	Data type	Description / Attributes
🕒	🔑	WorkOrderID	int	Primary key. Foreign key to WorkOrder.WorkOrderID. <b>References:</b> Production.WorkOrder
🕒	🔑	ProductID	int	Primary key. Foreign key to Product.ProductID.
🕒	🔑	OperationSequence	smallint	Primary key. Indicates the manufacturing process sequence.
🕒		LocationID	smallint	Manufacturing location where the part is processed. Foreign key to Location.LocationID. <b>References:</b> Production.Location
🕒		ScheduledStartDate	datetime	Planned manufacturing start date.
🕒		ScheduledEndDate	datetime	Planned manufacturing end date.
🕒		ActualStartDate	datetime	Actual start date. <b>Nullable</b>
🕒		ActualEndDate	datetime	Actual end date. <b>Nullable</b>

	Name	Data type	Description / Attributes
■	ActualResourceHrs	decimal(9, 4)	Number of manufacturing hours used. <b>Nullable</b>
■	PlannedCost	money	Estimated manufacturing cost.
■	ActualCost	money	Actual manufacturing cost. <b>Nullable</b>
■	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

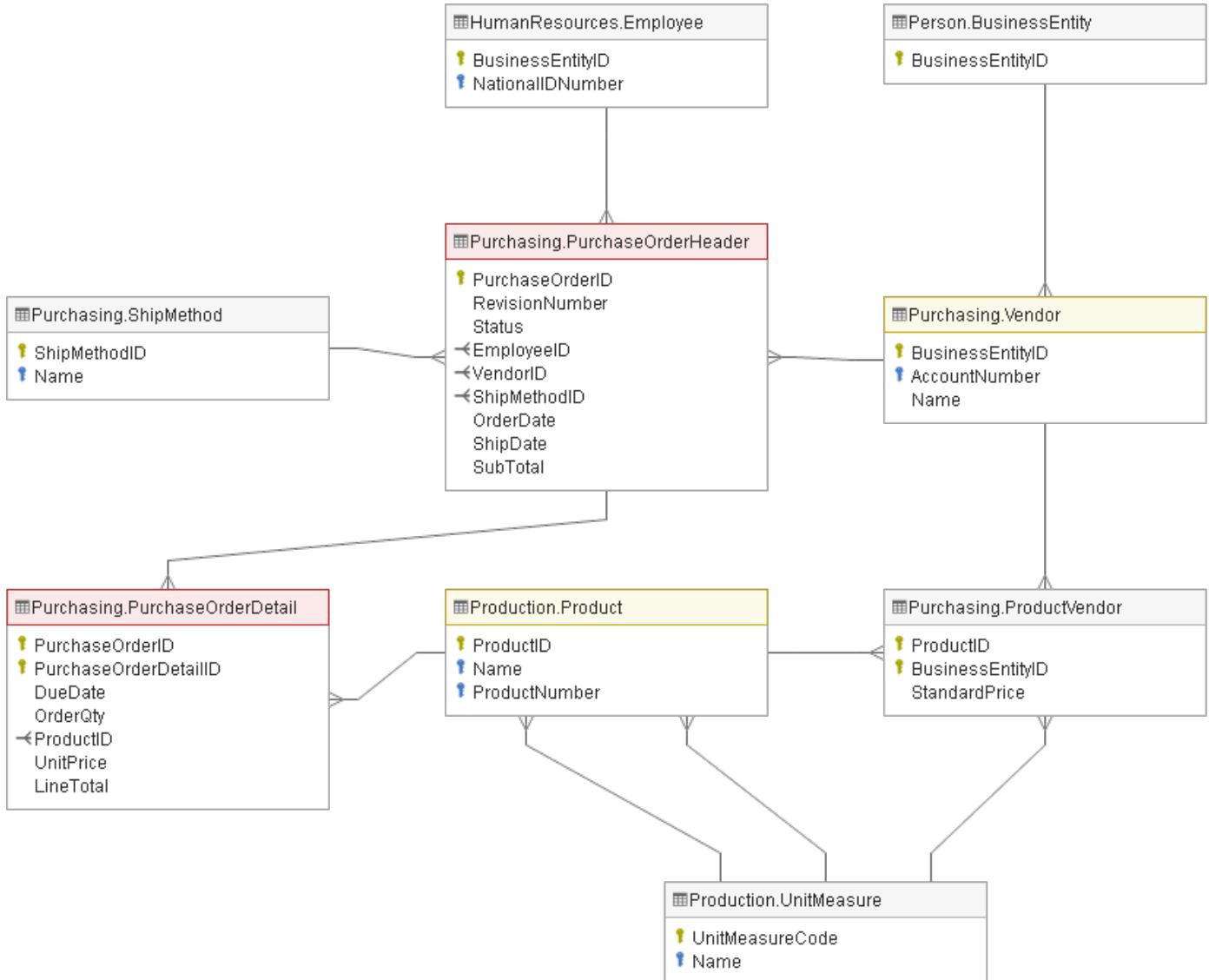
## Links to

	Table	Join	Title / Name / Description
→	Production.Location	<b>Production.WorkOrderRouting.Location</b> .LocationID = Production.Location.LocationID	FK_WorkOrderRouting_Location_LocationID Foreign key constraint referencing Location.LocationID.
→	Production.WorkOrder	<b>Production.WorkOrderRouting.WorkOrder</b> .WorkOrderID = Production.WorkOrder.WorkOrderID	FK_WorkOrderRouting_WorkOrder_WorkOrderID Foreign key constraint referencing WorkOrder.WorkOrderID.

## Unique keys

	Columns	Name / Description
🔑	WorkOrderID, ProductID, OperationSequence	PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence Primary key (clustered) constraint

## 1.6. Purchasing



Vendors from who parts and products are purchased.

### 1.6.1. Tables

#### 1.6.1.1. Table: Purchasing.ProductVendor

Cross-reference table mapping vendors with the products they supply.

##### Columns

	Name	Data type	Description / Attributes
PK	ProductID	int	Primary key. Foreign key to Product.ProductID. <b>References:</b> Production.Product
PK	BusinessEntityID	int	Primary key. Foreign key to Vendor.BusinessEntityID. <b>References:</b> Purchasing.Vendor
	AverageLeadTime	int	The average span of time (in days) between placing an order with the vendor and receiving the purchased product.
	StandardPrice	money	The vendor's usual selling price.
	LastReceiptCost	money	The selling price when last purchased. <b>Nullable</b>

	Name	Data type	Description / Attributes
>LastReceiptDate	LastReceiptDate	datetime	Date the product was last received by the vendor. <b>Nullable</b>
MinOrderQty	MinOrderQty	int	The maximum quantity that should be ordered.
MaxOrderQty	MaxOrderQty	int	The minimum quantity that should be ordered.
OnOrderQty	OnOrderQty	int	The quantity currently on order. <b>Nullable</b>
UnitMeasureCode	UnitMeasureCode	nchar(3)	The product's unit of measure. <b>References:</b> Production.UnitMeasure
ModifiedDate	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Purchasing.ProductVendor</b> .ProductID = Production.Product.ProductID	FK_ProductVendor_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Production.UnitMeasure	<b>Purchasing.ProductVendor</b> .UnitMeasureCode = Production.UnitMeasure.UnitMeasureCode	FK_ProductVendor_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
→	Purchasing.Vendor	<b>Purchasing.ProductVendor</b> .BusinessEntityID = Purchasing.Vendor.BusinessEntityID	FK_ProductVendor_Vendor_BusinessEntityID Foreign key constraint referencing Vendor.BusinessEntityID.

Unique keys

	Columns	Name / Description
🔑	ProductID, BusinessEntityID	PK_ProductVendor_ProductID_BusinessEntityID Primary key (clustered) constraint

### 1.6.1.2. Table: Purchasing.PurchaseOrderDetail

Individual products associated with a specific purchase order. See PurchaseOrderHeader.

Columns

	Name	Data type	Description / Attributes
🔑	PurchaseOrderID	int	Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID. <b>References:</b> Purchasing.PurchaseOrderHeader
🔑	PurchaseOrderDetailID	int	Primary key. One line number per purchased product. <b>Identity / Auto increment</b>
	DueDate	datetime	Date the product is expected to be received.
	OrderQty	smallint	Quantity ordered.
	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
	UnitPrice	money	Vendor's selling price of a single product.
	LineTotal	money	Per product subtotal. Computed as OrderQty * UnitPrice. <b>Computed:</b> isnull([OrderQty]*[UnitPrice],(0.00))
	ReceivedQty	decimal(8, 2)	Quantity actually received from the vendor.
	RejectedQty	decimal(8, 2)	Quantity rejected during inspection.

	Name	Data type	Description / Attributes
StockedQty	decimal(9, 2)		Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty. <b>Computed:</b> isnull([ReceivedQty]-[RejectedQty],(0.00))
ModifiedDate	datetime		Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
Production.Product	Purchasing.PurchaseOrderDetail.ProductID = Production.Product.ProductID		FK_PurchaseOrderDetail_Product_ProductID Foreign key constraint referencing Product.ProductID.
Purchasing.PurchaseOrderHeader	Purchasing.PurchaseOrderDetail.PurchaseOrderID = Purchasing.PurchaseOrderHeader.PurchaseOrderID		FK_PurchaseOrderDetail_PurchaseOrderHeader_PurchaseOrderID Foreign key constraint referencing PurchaseOrderHeader.PurchaseOrderID.

Unique keys

	Columns	Name / Description
PK	PurchaseOrderID, PurchaseOrderDetailID	PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID Primary key (clustered) constraint

Triggers

	Name	When	Description
⚡	iPurchaseOrderDetail	After Insert	AFTER INSERT trigger that inserts a row in the TransactionHistory table and updates the PurchaseOrderHeader.SubTotal column.
⚡	uPurchaseOrderDetail	After Update	AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in PurchaseOrderDetail and updates the PurchaseOrderHeader.SubTotal column.

### 1.6.1.3. Table: Purchasing.PurchaseOrderHeader

General purchase order information. See PurchaseOrderDetail.

Columns

	Name	Data type	Description / Attributes
PK	PurchaseOrderID	int	Primary key. <b>Identity / Auto increment</b>
PK	RevisionNumber	tinyint	Incremental number to track changes to the purchase order over time. <b>Default:</b> 0
PK	Status	tinyint	Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete <b>Default:</b> 1
PK	EmployeeID	int	Employee who created the purchase order. Foreign key to Employee.BusinessEntityID. <b>References:</b> HumanResources.Employee
PK	VendorID	int	Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntityID. <b>References:</b> Purchasing.Vendor
PK	ShipMethodID	int	Shipping method. Foreign key to ShipMethod.ShipMethodID. <b>References:</b> Purchasing.ShipMethod
PK	OrderDate	datetime	Purchase order creation date. <b>Default:</b> getdate()

	Name	Data type	Description / Attributes
■	ShipDate	datetime	Estimated shipment date from the vendor. <b>Nullable</b>
■	SubTotal	money	Purchase order subtotal. Computed as SUM(PurchaseOrderDetail.LineTotal)for the appropriate PurchaseOrderID. <b>Default:</b> 0.00
■	TaxAmt	money	Tax amount. <b>Default:</b> 0.00
■	Freight	money	Shipping cost. <b>Default:</b> 0.00
■	TotalDue	money	Total due to vendor. Computed as Subtotal + TaxAmt + Freight. <b>Computed:</b> isnull(([SubTotal]+[TaxAmt])+[Freight],0)
■	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	HumanResources.Employee	<b>Purchasing.PurchaseOrderHeader.EmployeeID</b> = HumanResources.Employee.BusinessEntityID	FK_PurchaseOrderHeader_Employee_EmployeeID Foreign key constraint referencing Employee.EmployeeID.
→	Purchasing.ShipMethod	<b>Purchasing.PurchaseOrderHeader.ShipMethodID</b> = Purchasing.ShipMethod.ShipMethodID	FK_PurchaseOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.ShipMethodID.
→	Purchasing.Vendor	<b>Purchasing.PurchaseOrderHeader.VendorID</b> = Purchasing.Vendor.BusinessEntityID	FK_PurchaseOrderHeader_Vendor_VendorID Foreign key constraint referencing Vendor.VendorID.

Linked from

	Table	Join	Title / Name / Description
←	Purchasing.PurchaseOrderDetail	<b>Purchasing.PurchaseOrderHeader.PurchaseOrderID</b> = Purchasing.PurchaseOrderDetail.PurchaseOrderID	FK_PurchaseOrderDetail_PurchaseOrderHeader_PurchaseOrderID Foreign key constraint referencing PurchaseOrderHeader.PurchaseOrderID.

Unique keys

	Columns	Name / Description
🔑	PurchaseOrderID	PK_PurchaseOrderHeader_PurchaseOrderID Primary key (clustered) constraint

Triggers

	Name	When	Description
⚡	uPurchaseOrderHeader	After Update	AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the PurchaseOrderHeader table.

#### 1.6.1.4. Table: Purchasing.ShipMethod

Shipping company lookup table.

Columns

	Name	Data type	Description / Attributes
■	ShipMethodID	int	Primary key for ShipMethod records. <b>Identity / Auto increment</b>

	Name	Data type	Description / Attributes
PK	Name	nvarchar(50)	Shipping company name.
PK	ShipBase	money	Minimum shipping charge. <b>Default:</b> 0.00
PK	ShipRate	money	Shipping charge per pound. <b>Default:</b> 0.00
PK	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
PK	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Linked from

	Table	Join	Title / Name / Description
→	Purchasing.PurchaseOrderHeader	<b>Purchasing.ShipMethod.ShipMethodID</b> = Purchasing.PurchaseOrderHeader.ShipMethodID	FK_PurchaseOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.ShipMethodID.
→	Sales.SalesOrderHeader	<b>Purchasing.ShipMethod.ShipMethodID</b> = Sales.SalesOrderHeader.ShipMethodID	FK_SalesOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.ShipMethodID.

#### Unique keys

	Columns	Name / Description
PK	ShipMethodID	PK_ShipMethod_ShipMethodID Primary key (clustered) constraint
PK	Name	AK_ShipMethod_Name Unique nonclustered index.
PK	rowguid	AK_ShipMethod_rowguid Unique nonclustered index. Used to support replication samples.

#### 1.6.1.5. Table: Purchasing.Vendor

Companies from whom Adventure Works Cycles purchases parts or other goods.

#### Columns

	Name	Data type	Description / Attributes
PK	BusinessEntityID	int	Primary key for Vendor records. Foreign key to BusinessEntity.BusinessEntityID <b>References:</b> Person.BusinessEntity
PK	AccountNumber	nvarchar(15)	Vendor account (identification) number.
PK	Name	nvarchar(50)	Company name.
PK	CreditRating	tinyint	1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average
PK	PreferredVendorStatus	bit	0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product. <b>Default:</b> 1
PK	ActiveFlag	bit	0 = Vendor no longer used. 1 = Vendor is actively used. <b>Default:</b> 1
PK	PurchasingWebServiceURL	nvarchar(1024)	Vendor URL. <b>Nullable</b>

		Name	Data type	Description / Attributes
🕒	ModifiedDate		datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
➤	Person.BusinessEntity	Purchasing.Vendor.BusinessEntityID = Person.BusinessEntity.BusinessEntityID	FK_Vendor_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID

Linked from

	Table	Join	Title / Name / Description
→	Purchasing.ProductVendor	Purchasing.Vendor.BusinessEntityID = Purchasing.ProductVendor.BusinessEntityID	FK_ProductVendor_Vendor_BusinessEntityID Foreign key constraint referencing Vendor.BusinessEntityID.
→	Purchasing.PurchaseOrderHeader	Purchasing.Vendor.BusinessEntityID = Purchasing.PurchaseOrderHeader.VendorID	FK_PurchaseOrderHeader_Vendor_VendorID Foreign key constraint referencing Vendor.VendorID.

Unique keys

	Columns	Name / Description	
🔑	BusinessEntityID	PK_Vendor_BusinessEntityID	Primary key (clustered) constraint
🔑	AccountNumber	AK_Vendor_AccountNumber	Unique nonclustered index.

Triggers

	Name	When	Description
⚡	dVendor	Instead Of Delete	INSTEAD OF DELETE trigger which keeps Vendors from being deleted.

## 1.6.2. Views

### 1.6.2.1. View: Purchasing.vVendorWithAddresses

Vendor (company) names and addresses .

Columns

	Name	Data type	Description / Attributes
🕒	BusinessEntityID	int	
🕒	Name	nvarchar(50)	
🕒	AddressType	nvarchar(50)	
🕒	AddressLine1	nvarchar(60)	
🕒	AddressLine2	nvarchar(60)	<b>Nullable</b>
🕒	City	nvarchar(30)	
🕒	StateProvinceName	nvarchar(50)	
🕒	PostalCode	nvarchar(15)	
🕒	CountryRegionName	nvarchar(50)	

### 1.6.2.2. View: Purchasing.vVendorWithContacts

Vendor (company) names and the names of vendor employees to contact.

#### Columns

	Name	Data type	Description / Attributes
BusinessEntityID	BusinessEntityID	int	
Name	Name	nvarchar(50)	
ContactType	ContactType	nvarchar(50)	
Title	Title	nvarchar(8)	<b>Nullable</b>
FirstName	FirstName	nvarchar(50)	
MiddleName	MiddleName	nvarchar(50)	<b>Nullable</b>
LastName	LastName	nvarchar(50)	
Suffix	Suffix	nvarchar(10)	<b>Nullable</b>
PhoneNumber	PhoneNumber	nvarchar(25)	<b>Nullable</b>
PhoneNumberType	PhoneNumberType	nvarchar(50)	<b>Nullable</b>
EmailAddress	EmailAddress	nvarchar(50)	<b>Nullable</b>
EmailPromotion	EmailPromotion	int	

### 1.6.3. Functions

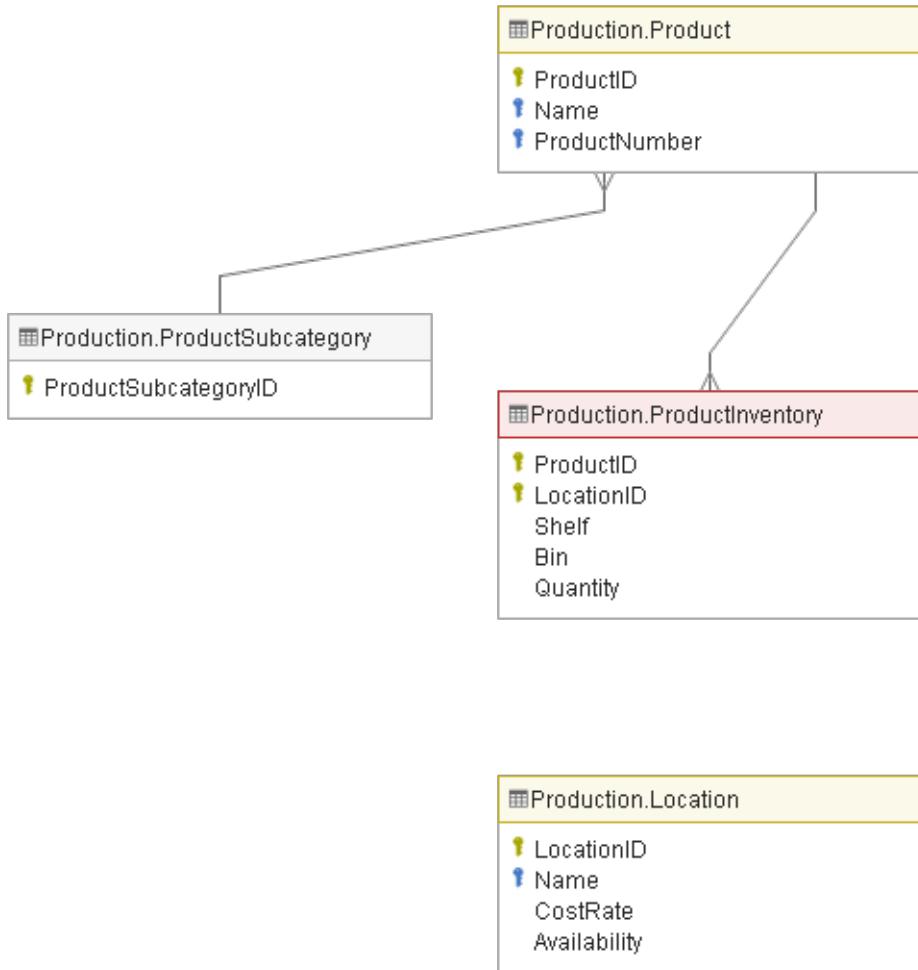
#### 1.6.3.1. Function: dbo.ufnGetPurchaseOrderStatusText

Scalar function returning the text representation of the Status column in the PurchaseOrderHeader table.

#### Input/Output

	Name	Data type	Description
*@	Returns	nvarchar(15)	
*@	Status	tinyint	Input parameter for the scalar function ufnGetPurchaseOrderStatusText. Enter a valid integer.

## 1.7. Inventory



Inventory management.

### 1.7.1. Tables

#### 1.7.1.1. Table: Production.Location

Product inventory and manufacturing locations.

Columns

		Name	Data type	Description / Attributes
1	2	LocationID	smallint	Primary key for Location records. <b>Identity / Auto increment</b>
1	3	Name	nvarchar(50)	Location description.
1	4	CostRate	smallmoney	Standard hourly cost of the manufacturing location. <b>Default:</b> 0.00
1	5	Availability	decimal(8, 2)	Work capacity (in hours) of the manufacturing location. <b>Default:</b> 0.00
1	6	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Production.ProductInventory	<b>Production.Location.LocationID</b> = Production.ProductInventory.LocationID	FK_ProductInventory_Location_LocationID Foreign key constraint referencing Location.LocationID.
→	Production.WorkOrderRouting	<b>Production.Location.LocationID</b> = Production.WorkOrderRouting.LocationID	FK_WorkOrderRouting_Location_LocationID Foreign key constraint referencing Location.LocationID.

## Unique keys

	Columns	Name / Description
🔑	LocationID	PK_Location_LocationID Primary key (clustered) constraint
🔑	Name	AK_Location_Name Unique nonclustered index.

## 1.7.1.2. Table: Production.ProductInventory

Product inventory information.

### Columns

	Name	Data type	Description / Attributes
☰	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
☰	LocationID	smallint	Inventory location identification number. Foreign key to Location.LocationID. <b>References:</b> Production.Location
☰	Shelf	nvarchar(10)	Storage compartment within an inventory location.
☰	Bin	tinyint	Storage container on a shelf in an inventory location.
☰	Quantity	smallint	Quantity of products in the inventory location. <b>Default:</b> 0
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

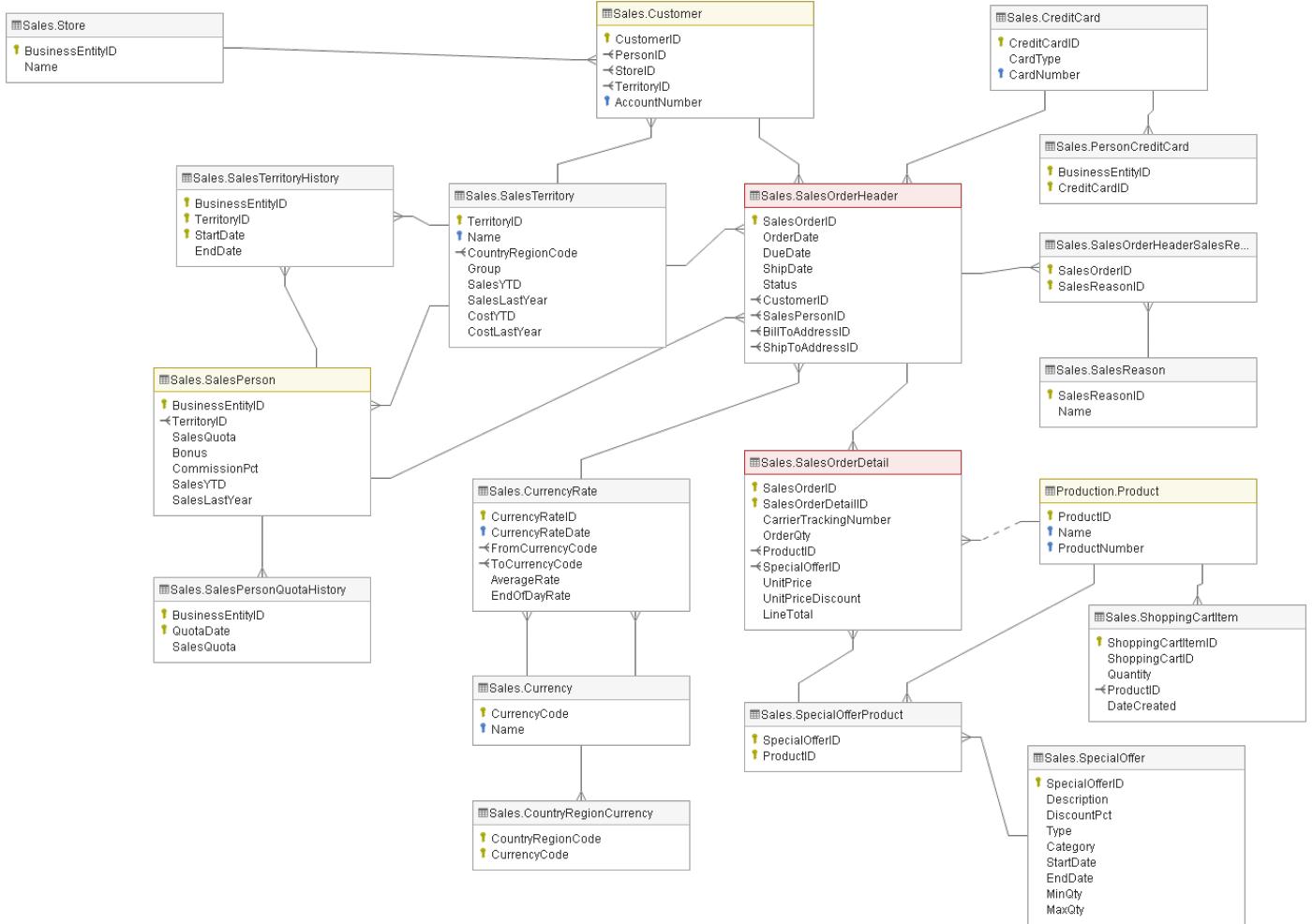
## Links to

	Table	Join	Title / Name / Description
→	Production.Location	<b>Production.ProductInventory.LocationID</b> = Production.Location.LocationID	FK_ProductInventory_Location_LocationID Foreign key constraint referencing Location.LocationID.
→	Production.Product	<b>Production.ProductInventory.ProductID</b> = Production.Product.ProductID	FK_ProductInventory_Product_ProductID Foreign key constraint referencing Product.ProductID.

## Unique keys

	Columns	Name / Description
🔑	ProductID, LocationID	PK_ProductInventory_ProductID_LocationID Primary key (clustered) constraint

## 1.8. Sales



Sales module holds information about shopping cart, sales orders, special offers and sales people.

### 1.8.1. Tables

#### 1.8.1.1. Table: Sales.CountryRegionCurrency

Cross-reference table mapping ISO currency codes to a country or region.

##### Columns

	Name	Data type	Description / Attributes
PK	CountryRegionCode	nvarchar(3)	ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode. <b>References:</b> Person.CountryRegion
PK	CurrencyCode	nchar(3)	ISO standard currency code. Foreign key to Currency.CurrencyCode. <b>References:</b> Sales.Currency
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

##### Links to

	Table	Join	Title / Name / Description
→	Person.CountryRegion	Sales.CountryRegionCurrency.CountryRegionCode = Person.CountryRegion.CountryRegionCode	FK_CountryRegionCurrency_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.

	Table	Join	Title / Name / Description
→	Sales.Currency	<b>Sales.CountryRegionCurrency.CurrencyCode</b> = Sales.Currency.CurrencyCode	FK_CountryRegionCurrency_Currency_CurrencyCode Foreign key constraint referencing Currency.CurrencyCode.

## Unique keys

	Columns	Name / Description
🔑	CountryRegionCode, CurrencyCode	PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode Primary key (clustered) constraint

## 1.8.1.2. Table: Sales.CreditCard

Customer credit card information.

### Columns

	Name	Data type	Description / Attributes
🔑	CreditCardID	int	Primary key for CreditCard records. <b>Identity / Auto increment</b>
☰	CardType	nvarchar(50)	Credit card name.
🔑	CardNumber	nvarchar(25)	Credit card number.
☰	ExpMonth	tinyint	Credit card expiration month.
☰	ExpYear	smallint	Credit card expiration year.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

### Linked from

	Table	Join	Title / Name / Description
←	Sales.PersonCreditCard	<b>Sales.CreditCard.CreditCardID</b> = Sales.PersonCreditCard.CreditCardID	FK_PersonCreditCard_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCardID.
←	Sales.SalesOrderHeader	<b>Sales.CreditCard.CreditCardID</b> = Sales.SalesOrderHeader.CreditCardID	FK_SalesOrderHeader_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCardID.

## Unique keys

	Columns	Name / Description
🔑	CreditCardID	PK_CreditCard_CreditCardID Primary key (clustered) constraint
🔑	CardNumber	AK_CreditCard_CardNumber Unique nonclustered index.

## 1.8.1.3. Table: Sales.Currency

Lookup table containing standard ISO currencies.

### Columns

	Name	Data type	Description / Attributes
🔑	CurrencyCode	nchar(3)	The ISO code for the Currency.
🔑	Name	nvarchar(50)	Currency name.
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Linked from

	Table	Join	Title / Name / Description
→	Sales.CountryRegionCurrency	<b>Sales.Currency.CurrencyCode</b> = Sales.CountryRegionCurrency.CurrencyCode	FK_CountryRegionCurrency_Currency_CurrencyCode Foreign key constraint referencing Currency.CurrencyCode.
→	Sales.CurrencyRate	<b>Sales.Currency.CurrencyCode</b> = Sales.CurrencyRate.FromCurrencyCode	FK_CurrencyRate_Currency_FromCurrencyCode Foreign key constraint referencing Currency.FromCurrencyCode.
→	Sales.CurrencyRate	<b>Sales.Currency.CurrencyCode</b> = Sales.CurrencyRate.ToCurrencyCode	FK_CurrencyRate_Currency_ToCurrencyCode Foreign key constraint referencing Currency.FromCurrencyCode.

## Unique keys

	Columns	Name / Description
🔑	CurrencyCode	PK_Currency_CurrencyCode Primary key (clustered) constraint
🔑	Name	AK_Currency_Name Unique nonclustered index.

## 1.8.1.4. Table: Sales.CurrencyRate

Currency exchange rates.

### Columns

	Name	Data type	Description / Attributes
█	CurrencyRateID	int	Primary key for CurrencyRate records. <b>Identity / Auto increment</b>
█	CurrencyRateDate	datetime	Date and time the exchange rate was obtained.
█	FromCurrencyCode	nchar(3)	Exchange rate was converted from this currency code. <b>References:</b> Sales.Currency
█	ToCurrencyCode	nchar(3)	Exchange rate was converted to this currency code. <b>References:</b> Sales.Currency
█	AverageRate	money	Average exchange rate for the day.
█	EndOfDayRate	money	Final exchange rate for the day.
█	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	Sales.Currency	<b>Sales.CurrencyRate.FromCurrencyCode</b> = Sales.Currency.CurrencyCode	FK_CurrencyRate_Currency_FromCurrencyCode Foreign key constraint referencing Currency.FromCurrencyCode.
→	Sales.Currency	<b>Sales.CurrencyRate.ToCurrencyCode</b> = Sales.Currency.CurrencyCode	FK_CurrencyRate_Currency_ToCurrencyCode Foreign key constraint referencing Currency.FromCurrencyCode.

## Linked from

	Table	Join	Title / Name / Description
→	Sales.SalesOrderHeader	<b>Sales.CurrencyRate.CurrencyRateID</b> = Sales.SalesOrderHeader.CurrencyRateID	FK_SalesOrderHeader_CurrencyRate_CurrencyRateID Foreign key constraint referencing CurrencyRate.CurrencyRateID.

## Unique keys

	Columns	Name / Description
🔑	CurrencyRateID	PK_CurrencyRate_CurrencyRateID Primary key (clustered) constraint
🔑	CurrencyRateDate, FromCurrencyCode, ToCurrencyCode	AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode Unique nonclustered index.

## 1.8.1.5. Table: Sales.Customer

Current customer information. Also see the Person and Store tables.

### Columns

	Name	Data type	Description / Attributes
☰	CustomerID	int	Primary key. <b>Identity / Auto increment</b>
☰	PersonID	int	Foreign key to Person.BusinessEntityID <b>Nullable</b> <b>References:</b> Person.Person
☰	StoreID	int	Foreign key to Store.BusinessEntityID <b>Nullable</b> <b>References:</b> Sales.Store
☰	TerritoryID	int	ID of the territory in which the customer is located. Foreign key to SalesTerritory.SalesTerritoryID. <b>Nullable</b> <b>References:</b> Sales.SalesTerritory
☰	AccountNumber	varchar(10)	Unique number identifying the customer assigned by the accounting system. <b>Computed:</b> isnull('AW'+[ufnLeadingZeros]([CustomerID]),")
☰	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

### Links to

	Table	Join	Title / Name / Description
→	Person.Person	<b>Sales.Customer.PersonID</b> = Person.Person.BusinessEntityID	FK_Customer_Person_PersonID Foreign key constraint referencing Person.BusinessEntityID.
→	Sales.SalesTerritory	<b>Sales.Customer.TerritoryID</b> = Sales.SalesTerritory.TerritoryID	FK_Customer_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
→	Sales.Store	<b>Sales.Customer.StoreID</b> = Sales.Store.BusinessEntityID	FK_Customer_Store_StoreID Foreign key constraint referencing Store.BusinessEntityID.

### Linked from

	Table	Join	Title / Name / Description
→	Sales.SalesOrderHeader	<b>Sales.Customer.CustomerID</b> = Sales.SalesOrderHeader.CustomerID	FK_SalesOrderHeader_Customer_CustomerID Foreign key constraint referencing Customer.CustomerID.

## Unique keys

	Columns	Name / Description
🔑	CustomerID	PK_Customer_CustomerID Primary key (clustered) constraint
🔑	AccountNumber	AK_Customer_AccountNumber Unique nonclustered index.
🔑	rowguid	AK_Customer_rowguid Unique nonclustered index. Used to support replication samples.

## 1.8.1.6. Table: Sales.PersonCreditCard

Cross-reference table mapping people to their credit card information in the CreditCard table.

### Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	Business entity identification number. Foreign key to Person.BusinessEntityID. <b>References:</b> Person.Person
☰	CreditCardID	int	Credit card identification number. Foreign key to CreditCard.CreditCardID. <b>References:</b> Sales.CreditCard
☰	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

### Links to

	Table	Join	Title / Name / Description
→	Sales.CreditCard	<b>Sales.PersonCreditCard.CreditCardID</b> = Sales.CreditCard.CreditCardID	FK_PersonCreditCard_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCardID.
→	Person.Person	<b>Sales.PersonCreditCard.BusinessEntityID</b> = Person.Person.BusinessEntityID	FK_PersonCreditCard_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntityID.

## Unique keys

	Columns	Name / Description
🔑	BusinessEntityID, CreditCardID	PK_PersonCreditCard_BusinessEntityID_CreditCardID Primary key (clustered) constraint

## 1.8.1.7. Table: Sales.SalesOrderDetail

Individual products associated with a specific sales order. See SalesOrderHeader.

### Columns

	Name	Data type	Description / Attributes
☰	SalesOrderID	int	Primary key. Foreign key to SalesOrderHeader.SalesOrderID. <b>References:</b> Sales.SalesOrderHeader
☰	SalesOrderDetailID	int	Primary key. One incremental unique number per product sold. <b>Identity / Auto increment</b>
☰	CarrierTrackingNumber	nvarchar(25)	Shipment tracking number supplied by the shipper. <b>Nullable</b>
☰	OrderQty	smallint	Quantity ordered per product.
☰	ProductID	int	Product sold to customer. Foreign key to Product.ProductID. <b>References:</b> Production.Product, Sales.SpecialOfferProduct

	Name	Data type	Description / Attributes
■	SpecialOfferID	int	Promotional code. Foreign key to SpecialOffer.SpecialOfferID. <b>References:</b> Sales.SpecialOfferProduct
■	UnitPrice	money	Selling price of a single product.
■	UnitPriceDiscount	money	Discount amount. <b>Default:</b> 0.0
■	LineTotal	numeric(38, 6)	Per product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * OrderQty. <b>Computed:</b> isnull(([UnitPrice]*((1.0)-[UnitPriceDiscount]))*[OrderQty],(0.0))
■ 	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
■	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Sales.SalesOrderDetail</b> .ProductID = Production.Product.ProductID	User-defined relation
→	Sales.SalesOrderHeader	<b>Sales.SalesOrderDetail</b> .SalesOrderID = Sales.SalesOrderHeader.SalesOrderID	FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.
→	Sales.SpecialOfferProduct	<b>Sales.SalesOrderDetail</b> .SpecialOfferID = Sales.SpecialOfferProduct.SpecialOfferID, <b>Sales.SalesOrderDetail</b> .ProductID = Sales.SpecialOfferProduct.ProductID	FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.

Unique keys

	Columns	Name / Description
🔑	SalesOrderID, SalesOrderDetailID	PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID Primary key (clustered) constraint
🔑	rowguid	AK_SalesOrderDetail_rowguid Unique nonclustered index. Used to support replication samples.

Triggers

	Name	When	Description
⚡	iduSalesOrderDetail	After Insert, Update, Delete	AFTER INSERT, DELETE, UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in SalesOrderDetail and updates the SalesOrderHeader.SubTotal column.

### 1.8.1.8. Table: Sales.SalesOrderHeader

General sales order information.

Columns

	Name	Data type	Description / Attributes
■ 	SalesOrderID	int	Primary key. <b>Identity / Auto increment</b>
■	RevisionNumber	tinyint	Incremental number to track changes to the sales order over time. <b>Default:</b> 0

	Name	Data type	Description / Attributes
■	OrderDate	datetime	Dates the sales order was created. <b>Default:</b> getdate()
■	DueDate	datetime	Date the order is due to the customer.
■	ShipDate	datetime	Date the order was shipped to the customer. <b>Nullable</b>
■	Status	tinyint	Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled <b>Default:</b> 1
■	OnlineOrderFlag	bit	0 = Order placed by sales person. 1 = Order placed online by customer. <b>Default:</b> 1
■	 SalesOrderNumber	nvarchar(25)	Unique sales order identification number. <b>Computed:</b> isnull(N'SO'+CONVERT([nvarchar](23),[SalesOrderID]),N'*** ERROR ***')
■	PurchaseOrderNumber	nvarchar(25)	Customer purchase order number reference. <b>Nullable</b>
■	AccountNumber	nvarchar(15)	Financial accounting number reference. <b>Nullable</b>
■	CustomerID	int	Customer identification number. Foreign key to Customer.BusinessEntityID. <b>References:</b> Sales.Customer
■	SalesPersonID	int	Sales person who created the sales order. Foreign key to SalesPerson.BusinessEntityID. <b>Nullable</b> <b>References:</b> Sales.SalesPerson
■	TerritoryID	int	Territory in which the sale was made. Foreign key to SalesTerritory.SalesTerritoryID. <b>Nullable</b> <b>References:</b> Sales.SalesTerritory
■	BillToAddressID	int	Customer billing address. Foreign key to Address.AddressID. <b>References:</b> Person.Address
■	ShipToAddressID	int	Customer shipping address. Foreign key to Address.AddressID. <b>References:</b> Person.Address
■	ShipMethodID	int	Shipping method. Foreign key to ShipMethod.ShipMethodID. <b>References:</b> Purchasing.ShipMethod
■	CreditCardID	int	Credit card identification number. Foreign key to CreditCard.CreditCardID. <b>Nullable</b> <b>References:</b> Sales.CreditCard
■	CreditCardApprovalCode	varchar(15)	Approval code provided by the credit card company. <b>Nullable</b>
■	CurrencyRateID	int	Currency exchange rate used. Foreign key to CurrencyRate.CurrencyRateID. <b>Nullable</b> <b>References:</b> Sales.CurrencyRate
■	SubTotal	money	Sales subtotal. Computed as SUM(SalesOrderDetail.LineTotal)for the appropriate SalesOrderID. <b>Default:</b> 0.00
■	TaxAmt	money	Tax amount. <b>Default:</b> 0.00
■	Freight	money	Shipping cost. <b>Default:</b> 0.00
■	TotalDue	money	Total due from customer. Computed as Subtotal + TaxAmt + Freight. <b>Computed:</b> isnull(([SubTotal]+[TaxAmt])+[Freight],0)

	Name	Data type	Description / Attributes
Comment	nvarchar(128)		Sales representative comments. <b>Nullable</b>
rowguid	uniqueidentifier		ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
ModifiedDate	datetime		Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
Person.Address	Sales.SalesOrderHeader.BillToAddressID = Person.Address.AddressID		FK_SalesOrderHeader_Address_BillToAddressID Foreign key constraint referencing Address.AddressID.
Person.Address	Sales.SalesOrderHeader.ShipToAddressID = Person.Address.AddressID		FK_SalesOrderHeader_Address_ShipToAddressID Foreign key constraint referencing Address.AddressID.
Sales.CreditCard	Sales.SalesOrderHeader.CreditCardID = Sales.CreditCard.CreditCardID		FK_SalesOrderHeader_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCardID.
Sales.CurrencyRate	Sales.SalesOrderHeader.CurrencyRateID = Sales.CurrencyRate.CurrencyRateID		FK_SalesOrderHeader_CurrencyRate_CurrencyRateID Foreign key constraint referencing CurrencyRate.CurrencyRateID.
Sales.Customer	Sales.SalesOrderHeader.CustomerID = Sales.Customer.CustomerID		FK_SalesOrderHeader_Customer_CustomerID Foreign key constraint referencing Customer.CustomerID.
Sales.SalesPerson	Sales.SalesOrderHeader.SalesPersonID = Sales.SalesPerson.BusinessEntityID		FK_SalesOrderHeader_SalesPerson_SalesPersonID Foreign key constraint referencing SalesPerson.SalesPersonID.
Sales.SalesTerritory	Sales.SalesOrderHeader.TerritoryID = Sales.SalesTerritory.TerritoryID		FK_SalesOrderHeader_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
Purchasing.ShipMethod	Sales.SalesOrderHeader.ShipMethodID = Purchasing.ShipMethod.ShipMethodID		FK_SalesOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.ShipMethodID.

## Linked from

	Table	Join	Title / Name / Description
Sales.SalesOrderDetail	Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID		FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.
Sales.SalesOrderHeaderSalesReason	Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderHeaderSalesReason.SalesOrderID		FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.SalesOrderID.

## Unique keys

	Columns	Name / Description
SalesOrderID	PK_SalesOrderHeader_SalesOrderID	Primary key (clustered) constraint
rowguid	AK_SalesOrderHeader_rowguid	Unique nonclustered index. Used to support replication samples.
SalesOrderNumber	AK_SalesOrderHeader_SalesOrderNumber	Unique nonclustered index.

## Triggers

	Name	When	Description
⚡	uSalesOrderHeader	After Update	AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the SalesOrderHeader table. Updates the SalesYTD column in the SalesPerson and SalesTerritory tables.

### 1.8.1.9. Table: Sales.SalesOrderHeaderSalesReason

Cross-reference table mapping sales orders to sales reason codes.

#### Columns

	Name	Data type	Description / Attributes
.PrimaryKey	SalesOrderID	int	Primary key. Foreign key to SalesOrderHeader.SalesOrderID. <b>References:</b> Sales.SalesOrderHeader
PrimaryKey	SalesReasonID	int	Primary key. Foreign key to SalesReason.SalesReasonID. <b>References:</b> Sales.SalesReason
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

#### Links to

	Table	Join	Title / Name / Description
→	Sales.SalesOrderHeader	<b>Sales.SalesOrderHeaderSalesReason.</b> SalesOrderID = Sales.SalesOrderHeader.SalesOrderID	FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.SalesOrderID.
→	Sales.SalesReason	<b>Sales.SalesOrderHeaderSalesReason.</b> SalesReasonID = Sales.SalesReason.SalesReasonID	FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID Foreign key constraint referencing SalesReason.SalesReasonID.

#### Unique keys

	Columns	Name / Description
PrimaryKey	SalesOrderID, SalesReasonID	PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID Primary key (clustered) constraint

### 1.8.1.10. Table: Sales.SalesPerson

Sales representative current information.

#### Columns

	Name	Data type	Description / Attributes
PrimaryKey	BusinessEntityID	int	Primary key for SalesPerson records. Foreign key to Employee.BusinessEntityID <b>References:</b> HumanResources.Employee
PrimaryKey	TerritoryID	int	Territory currently assigned to. Foreign key to SalesTerritory.SalesTerritoryID. <b>Nullable</b> <b>References:</b> Sales.SalesTerritory
PrimaryKey	SalesQuota	money	Projected yearly sales. <b>Nullable</b>
PrimaryKey	Bonus	money	Bonus due if quota is met. <b>Default:</b> 0.00
PrimaryKey	CommissionPct	smallmoney	Commission percent received per sale. <b>Default:</b> 0.00

	Name	Data type	Description / Attributes
[[table]]	SalesYTD	money	Sales total year to date. <b>Default:</b> 0.00
[[table]]	SalesLastYear	money	Sales total of previous year. <b>Default:</b> 0.00
[[table]] 	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
[[table]]	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

## Links to

	Table	Join	Title / Name / Description
→	HumanResources.Employee	<b>Sales.SalesPerson.BusinessEntityID</b> = HumanResources.Employee.BusinessEntityID	FK_SalesPerson_Employee_BusinessEntityID Foreign key constraint referencing Employee.EmployeeID.
→	Sales.SalesTerritory	<b>Sales.SalesPerson.TerritoryID</b> = Sales.SalesTerritory.TerritoryID	FK_SalesPerson_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.

## Linked from

	Table	Join	Title / Name / Description
←	Sales.SalesOrderHeader	<b>Sales.SalesPerson.BusinessEntityID</b> = Sales.SalesOrderHeader.SalesPersonID	FK_SalesOrderHeader_SalesPerson_SalesPersonID Foreign key constraint referencing SalesPerson.SalesPersonID.
←	Sales.SalesPersonQuotaHistory	<b>Sales.SalesPerson.BusinessEntityID</b> = Sales.SalesPersonQuotaHistory.BusinessEntityID	FK_SalesPersonQuotaHistory_SalesPerson_BusinessEntityID Foreign key constraint referencing SalesPerson.SalesPersonID.
←	Sales.SalesTerritoryHistory	<b>Sales.SalesPerson.BusinessEntityID</b> = Sales.SalesTerritoryHistory.BusinessEntityID	FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID Foreign key constraint referencing SalesPerson.SalesPersonID.
←	Sales.Store	<b>Sales.SalesPerson.BusinessEntityID</b> = Sales.Store.SalesPersonID	FK_Store_SalesPerson_SalesPersonID Foreign key constraint referencing SalesPerson.SalesPersonID

## Unique keys

	Columns	Name / Description
	BusinessEntityID	PK_SalesPerson_BusinessEntityID Primary key (clustered) constraint
	rowguid	AK_SalesPerson_rowguid Unique nonclustered index. Used to support replication samples.

## 1.8.1.11. Table: Sales.SalesPersonQuotaHistory

Sales performance tracking.

### Columns

	Name	Data type	Description / Attributes
[[table]] 	BusinessEntityID	int	Sales person identification number. Foreign key to SalesPerson.BusinessEntityID. <b>References:</b> Sales.SalesPerson
[[table]] 	QuotaDate	datetime	Sales quota date.
[[table]]	SalesQuota	money	Sales quota amount.

		Name	Data type	Description / Attributes
█	█	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Sales.SalesPerson	<b>Sales.SalesPersonQuotaHistory.</b> BusinessEntityID = Sales.SalesPerson.BusinessEntityID	FK_SalesPersonQuotaHistory_SalesPerson_BusinessEntityID Foreign key constraint referencing SalesPerson.SalesPersonID.

Unique keys

	Columns	Name / Description
█	BusinessEntityID, QuotaDate	PK_SalesPersonQuotaHistory_BusinessEntityID_QquotaDate Primary key (clustered) constraint
█	rowguid	AK_SalesPersonQuotaHistory_rowguid Unique nonclustered index. Used to support replication samples.

### 1.8.1.12. Table: Sales.SalesReason

Lookup table of customer purchase reasons.

Columns

		Name	Data type	Description / Attributes
█	█	SalesReasonID	int	Primary key for SalesReason records. <b>Identity / Auto increment</b>
█		Name	nvarchar(50)	Sales reason description.
█		ReasonType	nvarchar(50)	Category the sales reason belongs to.
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
→	Sales.SalesOrderHeaderSalesReason	<b>Sales.SalesReason.</b> SalesReasonID = Sales.SalesOrderHeaderSalesReason.SalesReasonID	FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID Foreign key constraint referencing SalesReason.SalesReasonID.

Unique keys

	Columns	Name / Description
█	SalesReasonID	PK_SalesReason_SalesReasonID Primary key (clustered) constraint

### 1.8.1.13. Table: Sales.SalesTaxRate

Tax rate lookup table.

Columns

		Name	Data type	Description / Attributes
█	█	SalesTaxRateID	int	Primary key for SalesTaxRate records. <b>Identity / Auto increment</b>

		Name	Data type	Description / Attributes
■	🔑	StateProvinceID	int	State, province, or country/region the sales tax applies to. <b>References:</b> Person.StateProvince
■	🔑	TaxType	tinyint	1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.
■		TaxRate	smallmoney	Tax rate amount. <b>Default:</b> 0.00
■		Name	nvarchar(50)	Tax rate description.
■	🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
■		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.StateProvince	<b>Sales.SalesTaxRate.StateProvinceID</b> = Person.StateProvince.StateProvinceID	FK_SalesTaxRate_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.StateProvinceID.

Unique keys

	Columns	Name / Description
🔑	SalesTaxRateID	PK_SalesTaxRate_SalesTaxRateID Primary key (clustered) constraint
🔑	rowguid	AK_SalesTaxRate_rowguid Unique nonclustered index. Used to support replication samples.
🔑	StateProvinceID, TaxType	AK_SalesTaxRate_StateProvinceID_TaxType Unique nonclustered index.

#### 1.8.1.14. Table: Sales.SalesTerritory

Sales territory lookup table.

Columns

		Name	Data type	Description / Attributes
■	🔑	TerritoryID	int	Primary key for SalesTerritory records. <b>Identity / Auto increment</b>
■	🔑	Name	nvarchar(50)	Sales territory description
■		CountryRegionCode	nvarchar(3)	ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode. <b>References:</b> Person.CountryRegion
■		Group	nvarchar(50)	Geographic area to which the sales territory belong.
■		SalesYTD	money	Sales in the territory year to date. <b>Default:</b> 0.00
■		SalesLastYear	money	Sales in the territory the previous year. <b>Default:</b> 0.00
■		CostYTD	money	Business costs in the territory year to date. <b>Default:</b> 0.00
■		CostLastYear	money	Business costs in the territory the previous year. <b>Default:</b> 0.00

		Name	Data type	Description / Attributes
█	█	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.CountryRegion	<b>Sales.SalesTerritory.CountryRegionCode</b> = Person.CountryRegion.CountryRegionCode	FK_SalesTerritory_CountryRegion_CountryRegionCode Foreign key constraint referencing CountryRegion.CountryRegionCode.

Linked from

	Table	Join	Title / Name / Description
←	Sales.Customer	<b>Sales.SalesTerritory.TerritoryID</b> = Sales.Customer.TerritoryID	FK_Customer_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
←	Sales.SalesOrderHeader	<b>Sales.SalesTerritory.TerritoryID</b> = Sales.SalesOrderHeader.TerritoryID	FK_SalesOrderHeader_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
←	Sales.SalesPerson	<b>Sales.SalesTerritory.TerritoryID</b> = Sales.SalesPerson.TerritoryID	FK_SalesPerson_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
←	Sales.SalesTerritoryHistory	<b>Sales.SalesTerritory.TerritoryID</b> = Sales.SalesTerritoryHistory.TerritoryID	FK_SalesTerritoryHistory_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.
←	Person.StateProvince	<b>Sales.SalesTerritory.TerritoryID</b> = Person.StateProvince.TerritoryID	FK_StateProvince_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.

Unique keys

	Columns	Name / Description
█	TerritoryID	PK_SalesTerritory_TerritoryID Primary key (clustered) constraint
█	Name	AK_SalesTerritory_Name Unique nonclustered index.
█	rowguid	AK_SalesTerritory_rowguid Unique nonclustered index. Used to support replication samples.

### 1.8.1.15. Table: Sales.SalesTerritoryHistory

Sales representative transfers to other sales territories.

Columns

		Name	Data type	Description / Attributes
█	█	BusinessEntityID	int	Primary key. The sales rep. Foreign key to SalesPerson.BusinessEntityID. <b>References:</b> Sales.SalesPerson
█	█	TerritoryID	int	Primary key. Territory identification number. Foreign key to SalesTerritory.SalesTerritoryID. <b>References:</b> Sales.SalesTerritory
█	█	StartDate	datetime	Primary key. Date the sales representative started work in the territory.
█		EndDate	datetime	Date the sales representative left work in the territory. <b>Nullable</b>

		Name	Data type	Description / Attributes
█	🔑	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
█		ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Sales.SalesPerson	<b>Sales.SalesTerritoryHistory.BusinessEntityID = Sales.SalesPerson.BusinessEntityID</b>	FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID Foreign key constraint referencing SalesPerson.SalesPersonID.
→	Sales.SalesTerritory	<b>Sales.SalesTerritoryHistory.TerritoryID = Sales.SalesTerritory.TerritoryID</b>	FK_SalesTerritoryHistory_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.

Unique keys

	Columns	Name / Description
🔑	BusinessEntityID, StartDate, TerritoryID	PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID Primary key (clustered) constraint
🔑	rowguid	AK_SalesTerritoryHistory_rowguid Unique nonclustered index. Used to support replication samples.

#### 1.8.1.16. Table: Sales.ShoppingCartItem

Contains online customer orders until the order is submitted or cancelled.

Columns

	Name	Data type	Description / Attributes
█	🔑 ShoppingCartItemID	int	Primary key for ShoppingCartItem records. <b>Identity / Auto increment</b>
█	ShoppingCartID	nvarchar(50)	Shopping cart identification number.
█	Quantity	int	Product quantity ordered. <b>Default:</b> 1
█	ProductID	int	Product ordered. Foreign key to Product.ProductID. <b>References:</b> Production.Product
█	DateCreated	datetime	Date the time the record was created. <b>Default:</b> getdate()
█	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Sales.ShoppingCartItem.ProductID = Production.Product.ProductID</b>	FK_ShoppingCartItem_Product_ProductID Foreign key constraint referencing Product.ProductID.

Unique keys

	Columns	Name / Description
🔑	ShoppingCartItemID	PK_ShoppingCartItem_ShoppingCartItemID Primary key (clustered) constraint

### 1.8.1.17. Table: Sales.SpecialOffer

Sale discounts lookup table.

Columns

	Name	Data type	Description / Attributes
PK	SpecialOfferID	int	Primary key for SpecialOffer records. <b>Identity / Auto increment</b>
	Description	nvarchar(255)	Discount description.
	DiscountPct	smallmoney	Discount percentage. <b>Default:</b> 0.00
	Type	nvarchar(50)	Discount type category.
	Category	nvarchar(50)	Group the discount applies to such as Reseller or Customer.
	StartDate	datetime	Discount start date.
	EndDate	datetime	Discount end date.
	MinQty	int	Minimum discount percent allowed. <b>Default:</b> 0
	MaxQty	int	Maximum discount percent allowed. <b>Nullable</b>
PK	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Linked from

	Table	Join	Title / Name / Description
FK	Sales.SpecialOfferProduct	Sales.SpecialOffer.SpecialOfferID = Sales.SpecialOfferProduct.SpecialOfferID	FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID Foreign key constraint referencing SpecialOffer.SpecialOfferID.

Unique keys

	Columns	Name / Description
PK	SpecialOfferID	PK_SpecialOffer_SpecialOfferID Primary key (clustered) constraint
AK	rowguid	AK_SpecialOffer_rowguid Unique nonclustered index. Used to support replication samples.

### 1.8.1.18. Table: Sales.SpecialOfferProduct

Cross-reference table mapping products to special offer discounts.

Columns

	Name	Data type	Description / Attributes
PK	SpecialOfferID	int	Primary key for SpecialOfferProduct records. <b>References:</b> Sales.SpecialOffer
PK	ProductID	int	Product identification number. Foreign key to Product.ProductID. <b>References:</b> Production.Product
PK	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()

	Name	Data type	Description / Attributes
>Last updated	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Production.Product	<b>Sales.SpecialOfferProduct</b> .ProductID = Production.Product.ProductID	FK_SpecialOfferProduct_Product_ProductID Foreign key constraint referencing Product.ProductID.
→	Sales.SpecialOffer	<b>Sales.SpecialOfferProduct</b> .SpecialOfferID = Sales.SpecialOffer.SpecialOfferID	FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID Foreign key constraint referencing SpecialOffer.SpecialOfferID.

Linked from

	Table	Join	Title / Name / Description
←	Sales.SalesOrderDetail	<b>Sales.SpecialOfferProduct</b> .SpecialOfferID = Sales.SalesOrderDetail.SpecialOfferID, <b>Sales.SpecialOfferProduct</b> .ProductID = Sales.SalesOrderDetail.ProductID	FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.

Unique keys

	Columns	Name / Description
PK	SpecialOfferID, ProductID	PK_SpecialOfferProduct_SpecialOfferID_ProductID Primary key (clustered) constraint
AK	rowguid	AK_SpecialOfferProduct_rowguid Unique nonclustered index. Used to support replication samples.

### 1.8.1.19. Table: Sales.Store

Customers (resellers) of Adventure Works products.

Columns

	Name	Data type	Description / Attributes
PK	BusinessEntityID	int	Primary key. Foreign key to Customer.BusinessEntityID. <b>References:</b> Person.BusinessEntity
	Name	nvarchar(50)	Name of the store.
	SalesPersonID	int	ID of the sales person assigned to the customer. Foreign key to SalesPerson.BusinessEntityID. <b>Nullable</b> <b>References:</b> Sales.SalesPerson
	Demographics	xml	Demographic information about the store such as the number of employees, annual sales and store type. <b>Nullable</b>
PK	rowguid	uniqueidentifier	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample. <b>Default:</b> newid()
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

Links to

	Table	Join	Title / Name / Description
→	Person.BusinessEntity	<b>Sales.Store</b> .BusinessEntityID = Person.BusinessEntity.BusinessEntityID	FK_Store_BusinessEntity_BusinessEntityID Foreign key constraint referencing BusinessEntity.BusinessEntityID

	Table	Join	Title / Name / Description
→	Sales.SalesPerson	<b>Sales.Store</b> .SalesPersonID = Sales.SalesPerson.BusinessEntityID	FK_Store_SalesPerson_SalesPersonID Foreign key constraint referencing SalesPerson.SalesPersonID

Linked from

	Table	Join	Title / Name / Description
←	Sales.Customer	<b>Sales.Store</b> .BusinessEntityID = Sales.Customer.StoreID	FK_Customer_Store_StoreID Foreign key constraint referencing Store.BusinessEntityID.

Unique keys

	Columns	Name / Description
🔑	BusinessEntityID	PK_Store_BusinessEntityID Primary key (clustered) constraint
🔑	rowguid	AK_Store_rowguid Unique nonclustered index. Used to support replication samples.

## 1.8.2. Views

### 1.8.2.1. View: Sales.vIndividualCustomer

Individual customers (names and addresses) that purchase Adventure Works Cycles products online.

Columns

	Name	Data type	Description / Attributes
☰	BusinessEntityID	int	
☰	Title	nvarchar(8)	<b>Nullable</b>
☰	FirstName	nvarchar(50)	
☰	MiddleName	nvarchar(50)	<b>Nullable</b>
☰	LastName	nvarchar(50)	
☰	Suffix	nvarchar(10)	<b>Nullable</b>
☰	PhoneNumber	nvarchar(25)	<b>Nullable</b>
☰	PhoneNumberType	nvarchar(50)	<b>Nullable</b>
☰	EmailAddress	nvarchar(50)	<b>Nullable</b>
☰	EmailPromotion	int	
☰	AddressType	nvarchar(50)	
☰	AddressLine1	nvarchar(60)	
☰	AddressLine2	nvarchar(60)	<b>Nullable</b>
☰	City	nvarchar(30)	
☰	StateProvinceName	nvarchar(50)	
☰	PostalCode	nvarchar(15)	
☰	CountryRegionName	nvarchar(50)	
☰	Demographics	xml	<b>Nullable</b>

### 1.8.2.2. View: Sales.vPersonDemographics

Displays the content from each element in the xml column Demographics for each customer in the Person.Person table.

## Columns

	Name	Data type	Description / Attributes
1	BusinessEntityID	int	
2	TotalPurchaseYTD	money	<b>Nullable</b>
3	DateFirstPurchase	datetime	<b>Nullable</b>
4	BirthDate	datetime	<b>Nullable</b>
5	MaritalStatus	nvarchar(1)	<b>Nullable</b>
6	YearlyIncome	nvarchar(30)	<b>Nullable</b>
7	Gender	nvarchar(1)	<b>Nullable</b>
8	TotalChildren	int	<b>Nullable</b>
9	NumberChildrenAtHome	int	<b>Nullable</b>
10	Education	nvarchar(30)	<b>Nullable</b>
11	Occupation	nvarchar(30)	<b>Nullable</b>
12	HomeOwnerFlag	bit	<b>Nullable</b>
13	NumberCarsOwned	int	<b>Nullable</b>

### 1.8.2.3. View: Sales.vSalesPerson

Sales representatives (names and addresses) and their sales-related information.

## Columns

	Name	Data type	Description / Attributes
1	BusinessEntityID	int	
2	Title	nvarchar(8)	<b>Nullable</b>
3	FirstName	nvarchar(50)	
4	MiddleName	nvarchar(50)	<b>Nullable</b>
5	LastName	nvarchar(50)	
6	Suffix	nvarchar(10)	<b>Nullable</b>
7	JobTitle	nvarchar(50)	
8	PhoneNumber	nvarchar(25)	<b>Nullable</b>
9	PhoneNumberType	nvarchar(50)	<b>Nullable</b>
10	EmailAddress	nvarchar(50)	<b>Nullable</b>
11	EmailPromotion	int	
12	AddressLine1	nvarchar(60)	
13	AddressLine2	nvarchar(60)	<b>Nullable</b>
14	City	nvarchar(30)	
15	StateProvinceName	nvarchar(50)	
16	PostalCode	nvarchar(15)	
17	CountryRegionName	nvarchar(50)	
18	TerritoryName	nvarchar(50)	<b>Nullable</b>
19	TerritoryGroup	nvarchar(50)	<b>Nullable</b>

	Name	Data type	Description / Attributes
■	SalesQuota	money	<b>Nullable</b>
■	SalesYTD	money	
■	SalesLastYear	money	

#### 1.8.2.4. View: Sales.vSalesPersonSalesByFiscalYears

Uses PIVOT to return aggregated sales information for each sales representative.

Columns

	Name	Data type	Description / Attributes
■	SalesPersonID	int	<b>Nullable</b>
■	FullName	nvarchar(152)	<b>Nullable</b>
■	JobTitle	nvarchar(50)	
■	SalesTerritory	nvarchar(50)	
■	2002	money	<b>Nullable</b>
■	2003	money	<b>Nullable</b>
■	2004	money	<b>Nullable</b>

#### 1.8.2.5. View: Sales.vStoreWithAddresses

Stores (including store addresses) that sell Adventure Works Cycles products to consumers.

Columns

	Name	Data type	Description / Attributes
■	BusinessEntityID	int	
■	Name	nvarchar(50)	
■	AddressType	nvarchar(50)	
■	AddressLine1	nvarchar(60)	
■	AddressLine2	nvarchar(60)	<b>Nullable</b>
■	City	nvarchar(30)	
■	StateProvinceName	nvarchar(50)	
■	PostalCode	nvarchar(15)	
■	CountryRegionName	nvarchar(50)	

#### 1.8.2.6. View: Sales.vStoreWithContacts

Stores (including store contacts) that sell Adventure Works Cycles products to consumers.

Columns

	Name	Data type	Description / Attributes
■	BusinessEntityID	int	
■	Name	nvarchar(50)	
■	ContactType	nvarchar(50)	
■	Title	nvarchar(8)	<b>Nullable</b>
■	FirstName	nvarchar(50)	

	Name	Data type	Description / Attributes
■	MiddleName	nvarchar(50)	<b>Nullable</b>
■	LastName	nvarchar(50)	
■	Suffix	nvarchar(10)	<b>Nullable</b>
■	PhoneNumber	nvarchar(25)	<b>Nullable</b>
■	PhoneNumberType	nvarchar(50)	<b>Nullable</b>
■	EmailAddress	nvarchar(50)	<b>Nullable</b>
■	EmailPromotion	int	

### 1.8.2.7. View: Sales.vStoreWithDemographics

Stores (including demographics) that sell Adventure Works Cycles products to consumers.

#### Columns

	Name	Data type	Description / Attributes
■	BusinessEntityID	int	
■	Name	nvarchar(50)	
■	AnnualSales	money	<b>Nullable</b>
■	AnnualRevenue	money	<b>Nullable</b>
■	BankName	nvarchar(50)	<b>Nullable</b>
■	BusinessType	nvarchar(5)	<b>Nullable</b>
■	YearOpened	int	<b>Nullable</b>
■	Specialty	nvarchar(50)	<b>Nullable</b>
■	SquareFeet	int	<b>Nullable</b>
■	Brands	nvarchar(30)	<b>Nullable</b>
■	Internet	nvarchar(30)	<b>Nullable</b>
■	NumberEmployees	int	<b>Nullable</b>

### 1.8.3. Functions

#### 1.8.3.1. Function: dbo.ufnGetAccountingEndDate

Scalar function used in the uSalesOrderHeader trigger to set the starting account date.

#### Input/Output

	Name	Data type	Description
*@	Returns	datetime	

#### 1.8.3.2. Function: dbo.ufnGetAccountingStartDate

Scalar function used in the uSalesOrderHeader trigger to set the ending account date.

#### Input/Output

	Name	Data type	Description
*@	Returns	datetime	

### 1.8.3.3. Function: dbo.ufnGetSalesOrderStatusText

Scalar function returning the text representation of the Status column in the SalesOrderHeader table.

#### Input/Output

	Name	Data type	Description
*@	Returns	nvarchar(15)	
>@	Status	tinyint	Input parameter for the scalar function ufnGetSalesOrderStatusText. Enter a valid integer.

### 1.8.3.4. Function: dbo.ufnLeadingZeros

Scalar function used by the Sales.Customer table to help set the account number.

#### Input/Output

	Name	Data type	Description
*@	Returns	varchar(8)	
>@	Value	int	Input parameter for the scalar function ufnLeadingZeros. Enter a valid integer.

## 1.9. Admin

### 1.9.1. Tables

#### 1.9.1.1. Table: dbo.AWBuildVersion

Current version number of the AdventureWorks 2012 sample database.

##### Columns

	Name	Data type	Description / Attributes
PK	SystemInformationID	tinyint	Primary key for AWBuildVersion records. <b>Identity / Auto increment</b>
	Database Version	nvarchar(25)	Version number of the database in 9.yy.mm.dd.00 format.
	VersionDate	datetime	Date and time the record was last updated.
	ModifiedDate	datetime	Date and time the record was last updated. <b>Default:</b> getdate()

##### Unique keys

	Columns	Name / Description
PK	SystemInformationID	PK_AWBuildVersion_SystemInformationID Primary key (clustered) constraint

#### 1.9.1.2. Table: dbo.DatabaseLog

Audit table tracking all DDL changes made to the AdventureWorks database. Data is captured by the database trigger `ddlDatabaseTriggerLog`.

##### Columns

	Name	Data type	Description / Attributes
PK	DatabaseLogID	int	Primary key for DatabaseLog records. <b>Identity / Auto increment</b>
	PostTime	datetime	The date and time the DDL change occurred.
	DatabaseUser	nvarchar(128)	The user who implemented the DDL change.
	Event	nvarchar(128)	The type of DDL statement that was executed.
	Schema	nvarchar(128)	The schema to which the changed object belongs. <b>Nullable</b>
	Object	nvarchar(128)	The object that was changed by the DDL statement. <b>Nullable</b>
	TSQL	nvarchar(MAX)	The exact Transact-SQL statement that was executed.
	XmlEvent	xml	The raw XML data generated by database trigger.

##### Unique keys

	Columns	Name / Description
PK	DatabaseLogID	PK_DatabaseLog_DatabaseLogID Primary key (nonclustered) constraint

#### 1.9.1.3. Table: dbo.ErrorLog

Audit table tracking errors in the the AdventureWorks database that are caught by the CATCH block of a TRY...CATCH construct. Data is inserted by stored procedure `dbo.usp.LogError` when it is executed from inside the CATCH block of a TRY...CATCH construct.

## Columns

	Name	Data type	Description / Attributes
	ErrorLogID	int	Primary key for ErrorLog records. <b>Identity / Auto increment</b>
	ErrorTime	datetime	The date and time at which the error occurred. <b>Default:</b> getdate()
	UserName	nvarchar(128)	The user who executed the batch in which the error occurred.
	ErrorNumber	int	The error number of the error that occurred.
	ErrorSeverity	int	The severity of the error that occurred. <b>Nullable</b>
	ErrorState	int	The state number of the error that occurred. <b>Nullable</b>
	ErrorProcedure	nvarchar(126)	The name of the stored procedure or trigger where the error occurred. <b>Nullable</b>
	ErrorLine	int	The line number at which the error occurred. <b>Nullable</b>
	ErrorMessage	nvarchar(4000)	The message text of the error that occurred.

## Unique keys

	Columns	Name / Description
	ErrorLogID	<b>PK_ErrorLog_ErrorLogID</b> Primary key (clustered) constraint

## 1.9.2. Procedures

### 1.9.2.1. Procedure: dbo.usp.LogError

Logs error information in the ErrorLog table about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.

#### Input/Output

	Name	Data type	Description
	ErrorLogID	int	Output parameter for the stored procedure usp.LogError. Contains the ErrorLogID value corresponding to the row inserted by usp.LogError in the ErrorLog table.

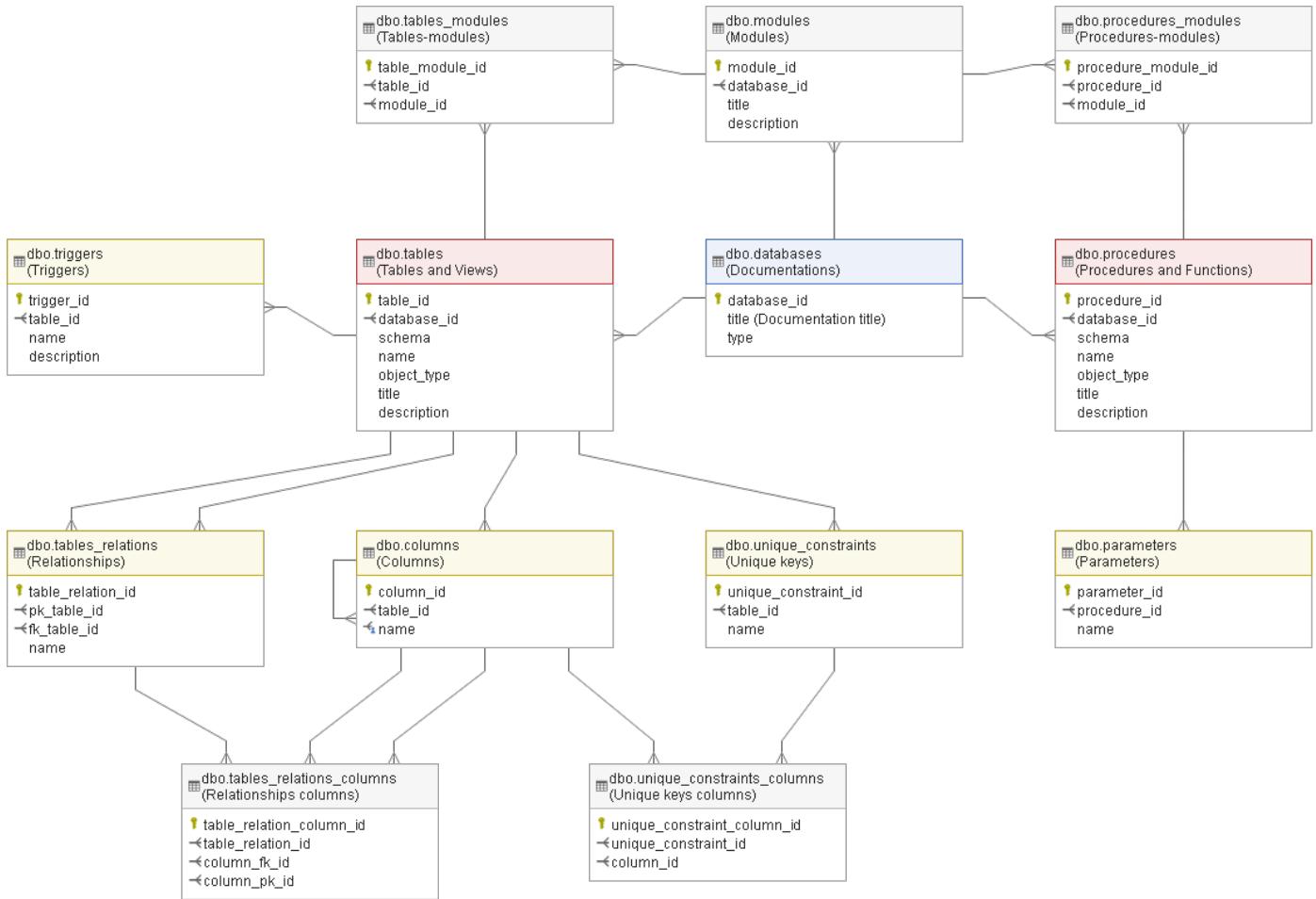
### 1.9.2.2. Procedure: dbo.uspPrintError

Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.

## 2. Dataedo repository

Dataedo 8.0 repository database documentation. Dataedo repository is SQL Server database that holds all imported database schemas and user created descriptions.

## 2.1. Schema and Metadata



All documented database objects are stored in Dataedo repository.

Distinct database documentations are stored in **databases** table which holds their title and DBMS connection details.

### Object status

Tables, views, columns, procedures, functions and parameters can be in one of two states identified by *status* column:

- active ('A') - object exists in source database or
- deleted ('B') - object has been deleted from source database. It is still visible in Dataedo Editor but is not included in exported documentation.

When object gets deleted with editor it gets physically deleted from Dataedo tables. There is no way to undelete it.

### Standard metadata columns

Every table has following metadata columns that describe who and when created and last updated data in repository:

- `creation_date` - timestamp of row's creation in Dataedo repository
- `created_by` - Dataedo user name that created the row in Dataedo repository
- `last_modification_date` - timestamp of row's last update in Dataedo repository
- `modified_by` - Dataedo user name that last updated the row in Dataedo repository

### DBMS import columns

Tables, views, procedures and functions have following metadata columns that facilitate change tracking and database schema updates:

- `dbms_creation_date` - timestamp of object creation in source database
- `dbms_last_modification_date` - timestamp of last object modification in source database
- `synchronization_date` - timestamp of last object update/import from source database
- `synchronized_by` - name of user that last updated/imported object from source database

## Object source

Objects representing data structures (tables, columns, relations etc) can have following sources identified by *source* column:

- 'DBMS' - object was imported from database schema,
- 'USER' - object was created by user in repository with Dataedo Editor and does not exist in database schema.

### 2.1.1. Tables

#### 2.1.1.1. Table: dbo.columns (Columns)

Holds tables' and views' columns.

##### Columns

	Name	Data type	Description / Attributes
	column_id	int	Row id <b>Identity / Auto increment</b>
	table_id	int	Table to which column belongs <b>References:</b> dbo.tables
	name	nvarchar(250)	Column name <b>References:</b> virtual.column_name_mask_match
	ordinal_position	int	Position of column in table in DBMS <b>Nullable</b>
	sort	int	Alternative column sorting defined by user in Dataedo <b>Default:</b> 99999
	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> N'DBMS'
	status	char(1)	'D' if column was deleted in DBMS, 'A' otherwise. Deleted columns aren't exported to documentation. <b>Default:</b> 'A'
	parent_id	int	Parent column for hierarchical data structures <b>Nullable</b> <b>References:</b> dbo.columns
	path	nvarchar(MAX)	Column path for hierarchical structures, e.g. field author.address.street will have street as name and author.address as path <b>Nullable</b>
	level	int	Hierarchy level for hierarchical structures. Regular "flat" structures such as tables have 1. <b>Default:</b> 1
	item_type	nvarchar(100)	Type of the field/column. Default one is COLUMN. Other types depend on data sources and are the following: ATTRIBUTE, COLUMN, COMPUTED, USER_TYPE, COLUMN_FAMILY, OBJECT, ARRAY, MAP, SET, OBJECT_ARRAY, ID, SYSTEM, CUSTM, HIERARCHY, MEASURE, MEASURE_GROUP, CALCULATED_MEASURE, KPI, DIMENSION <b>Default:</b> 'COLUMN'
	title	nvarchar(250)	Column title/alias in Dataedo repository <b>Nullable</b>
	description	nvarchar(MAX)	Column description in Dataedo. Description is imported from database at first import and during schema update, but only if it was empty <b>Nullable</b>
	datatype	nvarchar(250)	Column data type <b>Nullable</b>

	Name	Data type	Description / Attributes
■	data_length	nvarchar(50)	Data length/precision as text. Depending on data type it can have different formats, e.g. "100", "5,4", "MAX" etc. Null if parameter has no length/precision defined <b>Nullable</b>
■	nullable	bit	1 if column is nullable, 0 otherwise <b>Default:</b> 1
■	default_value	nvarchar(MAX)	Formula of column defalut value. NULL if field has not default value specification <b>Nullable</b>
■	is_identity	bit	Is column an identity/auto incremented column. 1 - identity/auto increment, 0 - otherwise <b>Default:</b> 0
■	is_computed	bit	Is column a computed column. 1 - computed column, 0 - regular column. If column is computed, you should expect that calculation formula is imported to computed_formula field <b>Default:</b> 0
■	computed_formula	nvarchar(MAX)	Calculation formula for computed columns. NULL for regular fields <b>Nullable</b>
■	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
■	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
■	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
■	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field17		nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18		nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19		nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20		nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21		nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22		nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23		nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24		nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25		nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26		nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27		nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28		nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29		nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30		nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31		nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32		nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33		nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34		nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35		nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36		nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37		nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38		nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39		nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40		nvarchar(MAX)	Custom field 40 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57	nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58	nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59	nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60	nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61	nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62	nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63	nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64	nvarchar(MAX)	Custom field 64 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	primary_key (Deprecated)	char(1)	Deprecated <b>Default:</b> 0
■	default_def (Deprecated)	nvarchar(4000)	Deprecated <b>Nullable</b>
■	identity_def (Deprecated)	nvarchar(50)	Deprecated <b>Nullable</b>
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates

## Links to

	Table	Join	Title / Name / Description
→	virtual.column_name_mask_match	<b>dbo.columns.name</b> = virtual.column_name_mask_match.col umn_name	fk_column_name_mask_match_columns
→	dbo.columns (Columns)	<b>dbo.columns.parent_id</b> = dbo.columns.column_id	FK_columns_column_parent_columns
→	dbo.schema_updates (Imports/Updates)	<b>dbo.columns.update_id</b> = dbo.schema_updates.update_id	fk_columns_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.columns.table_id</b> = dbo.tables.table_id	FK_column_table

## Linked from

	Table	Join	Title / Name / Description
←	dbo.columns (Columns)	<b>dbo.columns.column_id</b> = dbo.columns.parent_id	FK_columns_column_parent_columns
←	dbo.columns_changes (Columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.columns_changes.column_id	fk_columns_changes_columns
←	dbo.erd_nodes_columns (ERD entities columns)	<b>dbo.columns.column_id</b> = dbo.erd_nodes_columns.column_id	FK_erd_nodes_columns_columns
←	dbo.glossary_mappings (Term-Data links)	<b>dbo.columns.column_id</b> = dbo.glossary_mappings.element_id	fk_columns_glossary_mappings
←	dbo.tables_relations_columns (Relationships columns)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns.column_fk_id	FK_tables_relations_columns_column_fk
←	dbo.tables_relations_columns (Relationships columns)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns.column_pk_id	FK_tables_relations_columns_column_pk
←	dbo.tables_relations_columns_cha ges (Relationships columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns_changes .column_fk_id	fk_tables_relations_columns_changes_columns
←	dbo.tables_relations_columns_cha ges (Relationships columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns_changes .column_pk_id	fk_tables_relations_columns_changes_columns
←	dbo.tables_relations_columns_cha ges (Relationships columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns_changes .before_column_fk_id	fk_tables_relations_columns_changes_columns
←	dbo.tables_relations_columns_cha ges (Relationships columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.tables_relations_columns_changes .before_column_pk_id	fk_tables_relations_columns_changes_columns
←	dbo.unique_constraints_columns (Unique keys columns)	<b>dbo.columns.column_id</b> = dbo.unique_constraints_columns.col umn_id	FK_unique_constraints_columns_columns
←	dbo.unique_constraints_columns_c hanges (Unique keys columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.unique_constraints_columns_cha ges.column_id	fk_unique_constraints_columns_changes_columns
←	dbo.unique_constraints_columns_c hanges (Unique keys columns - Schema changes)	<b>dbo.columns.column_id</b> = dbo.unique_constraints_columns_cha ges.before_column_id	fk_unique_constraints_columns_changes_columns

## Unique keys

	Columns	Name / Description
🔑	column_id	PK_columns Table primary key

## Triggers

	Name	When	Description
⚡	trg_columns_change_track_insert	After Insert	Schema change tracking trigger
⚡	trg_columns_change_track_update	After Update	Schema change tracking trigger
⚡	trg_columns_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns.

### 2.1.1.2. Table: dbo.databases (Documentations)

Holds Dataedo documentations and their connection details.

- ▶  AdventureWorks
- ▶  Dataedo repository

## Columns

	Name	Data type	Description / Attributes
☰	🔑 database_id	int	Row id <b>Identity / Auto increment</b>
☰	guid	uniqueidentifier	Unique documentation identifier, unique across repositories <b>Default:</b> newsequentialid()
☰	title (Documentation title)	nvarchar(250)	Documentation title
☰	description	nvarchar(MAX)	Documentation text <b>Nullable</b>
☰	description_plain	nvarchar(MAX)	Holds description in plain text only, without HTML tags <b>Nullable</b>
☰	description_search	nvarchar(MAX)	Holds description in plain text and lower case. Column used for searching. <b>Nullable</b>
☰	type	nvarchar(100)	Type of documentation: - for DBMS its the type: ORACLE, SQLSERVER, etc. - MANUAL for manual documentations - BUSINESS_GLOSSARY for business glossary <b>Nullable</b>
☰	name (Database name)	nvarchar(MAX)	Database/schema name <b>Nullable</b>
☰	host	nvarchar(1024)	Host name/address <b>Nullable</b>
☰	service_name	nvarchar(100)	Service name for Oracle connection <b>Nullable</b>
☰	network_alias	nvarchar(100)	Network alias for Oracle TNS connection <b>Nullable</b>
☰	port	int	Port for Oracle connection <b>Nullable</b>
☰	windows_authentication	bit	Used for SQL Server: 1 if Windows authentication, 0 for SQL Server authentication <b>Nullable</b>

	Name	Data type	Description / Attributes
■	different_schema	bit	Used for Oracle: 1 if user selected schema different than user, 0 otherwise <b>Nullable</b>
■	multiple_schemas	bit	Used for Oracle: 1 if multiple schemas were selected <b>Nullable</b>
■	connection_type	nvarchar(100)	Used for Oracle: connection type: BASIC for host/port/service name, TNS for network alias <b>Nullable</b>
■	user	nvarchar(1024)	Username used to connect to database <b>Nullable</b>
■	password	nvarchar(MAX)	Encrypted password, null if not saved <b>Nullable</b>
■	dbms_version	nvarchar(500)	DBMS version as returned by DBMS at the time of last connection <b>Nullable</b>
■	filter	nvarchar(MAX)	Import filter definition saved in XML format <b>Nullable</b>
■	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
■	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
■	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
■	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17	nvarchar(MAX)	Custom field 17 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field18	nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19	nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20	nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21	nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22	nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23	nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24	nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25	nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26	nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27	nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28	nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29	nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30	nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31	nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34	nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35	nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()

	Name	Data type	Description / Attributes
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	ssl_key_path	nvarchar(500)	Path of the file that contains X.509 key <b>Nullable</b>
■	ssl_cert_path	nvarchar(500)	Path of the file that contains X.509 certificate <b>Nullable</b>
■	ssl_ca_path	nvarchar(500)	Path of the file that contains trusted SSL Certificate Authority certificate <b>Nullable</b>
■	ssl_cipher	nvarchar(MAX)	List of permitted ciphers for connection encryption <b>Nullable</b>
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field57		nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58		nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59		nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60		nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61		nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62		nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63		nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field81	nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82	nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83	nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84	nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85	nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86	nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87	nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88	nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>
■	instance_identifier	nvarchar(50)	Identifies which parameters to use for connection: SERVICE_NAME - service_name, SID - oracle_sid <b>Nullable</b>
■	oracle_sid	nvarchar(50)	SID for Oracle connection <b>Nullable</b>

## Linked from

	Table	Join	Title / Name / Description
→	dbo.columns_changes (Columns - Schema changes)	<b>dbo.databases.database_id = dbo.columns_changes.database_id</b>	fk_columns_changes_databases

	Table	Join	Title / Name / Description
→	dbo.dependencies_descriptions (Dependencies descriptions)	<b>dbo.databases.database_id</b> = dbo.dependencies_descriptions.database_id	FK_dependencies_descriptions_databases
→	dbo.documentation_custom_fields (Documentation custom fields)	<b>dbo.databases.database_id</b> = dbo.documentation_custom_fields.database_id	FK_documentation_custom_fields_databases
→	dbo.glossary_terms (Business Glossary Terms)	<b>dbo.databases.database_id</b> = dbo.glossary_terms.database_id	FK_glossary_terms_database
→	dbo.ignored_objects (Ignored objects)	<b>dbo.databases.database_id</b> = dbo.ignored_objects.database_id	FK_ignored_objects_databases
→	dbo.modules (Modules)	<b>dbo.databases.database_id</b> = dbo.modules.database_id	FK_modules_database
→	dbo.parameters_changes (Parameters - Schema changes)	<b>dbo.databases.database_id</b> = dbo.parameters_changes.database_id	fk_parameters_changes_databases
→	dbo.procedures (Procedures and Functions)	<b>dbo.databases.database_id</b> = dbo.procedures.database_id	FK_procedures_database
→	dbo.procedures_changes (Procedures and Functions - Schema changes)	<b>dbo.databases.database_id</b> = dbo.procedures_changes.database_id	fk_procedures_changes_databases
→	dbo.schema_updates (Imports/Updates)	<b>dbo.databases.database_id</b> = dbo.schema_updates.database_id	fk_schema_updates_databases
→	dbo.tables (Tables and Views)	<b>dbo.databases.database_id</b> = dbo.tables.database_id	FK_tables_databases
→	dbo.tables_changes (Tables and Views - Schema changes)	<b>dbo.databases.database_id</b> = dbo.tables_changes.database_id	fk_tables_changes_databases
→	dbo.tables_relations_changes (Relationships - Schema changes)	<b>dbo.databases.database_id</b> = dbo.tables_relations_changes.database_id	fk_tables_relations_changes_databases
→	dbo.triggers_changes (Triggers - Schema changes)	<b>dbo.databases.database_id</b> = dbo.triggers_changes.database_id	fk_triggers_changes_databases
→	dbo.unique_constraints_changes (Unique keys - Schema changes)	<b>dbo.databases.database_id</b> = dbo.unique_constraints_changes.database_id	fk_unique_constraints_changes_databases
→	dbo.user_connections (User connection details)	<b>dbo.databases.database_id</b> = dbo.user_connections.database_id	FK_user_connections_databases

## Unique keys

	Columns	Name / Description
🔑	database_id	PK_databases Table primary key

## Triggers

	Name	When	Description
⚡	trg_databases_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns.

### 2.1.1.3. Table: dbo.modules (Modules)

Database modules defined with Dataedo. Modules exist solely in Dataedo documentation repository and are used for logical grouping of database objects.

## Columns

	Name	Data type	Description / Attributes
💡	module_id	int	Row id <b>Identity / Auto increment</b>
💡	database_id	int	Documentation module is part of <b>References:</b> dbo.databases
💡	ordinal_position	int	Position of module in a list. By default null. Value assigned for all when sorted. If null then ordered by title. <b>Nullable</b>
💡	title	nvarchar(250)	Module name/title
💡	description	nvarchar(MAX)	Module introduction to be included in documentation <b>Nullable</b>
💡	description_plain	nvarchar(MAX)	Holds description in plain text only, without HTML tags <b>Nullable</b>
💡	description_search	nvarchar(MAX)	Holds description in plain text and lower case. Column used for searching. <b>Nullable</b>
💡	erd_link_style	nvarchar(25)	Default link ERD style. See erd_links.link_style for details. <b>Nullable</b> <b>Default:</b> N'STRAIGHT'
💡	erd_show_types	bit	Flag indicating if entities in ERD include data types. 1 - show, 0 - don't show. Default is 0. <b>Nullable</b> <b>Default:</b> 0
💡	display_documentation_name_mod e	nvarchar(25)	Flag defining whether documentation title should be shown for entities: EXTERNAL_ENTITIES_ONLY - show only for the ones from external documentations, default behaviour when null ALL_ENTITIES - show for all entities NO_ENTITIES - show for no entities <b>Nullable</b> <b>Default:</b> N'EXTERNAL_ENTITIES_ONLY'
💡	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
💡	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
💡	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
💡	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
💡	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
💡	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
💡	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
💡	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
💡	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
💡	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
💡	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field12		nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13		nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14		nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15		nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16		nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17		nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18		nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19		nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20		nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21		nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22		nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23		nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24		nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25		nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26		nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27		nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28		nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29		nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30		nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31		nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32		nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33		nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34		nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35		nvarchar(MAX)	Custom field 35 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field55		nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56		nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57		nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58		nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59		nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60		nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61		nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62		nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63		nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field79	nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80	nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81	nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82	nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83	nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84	nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85	nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86	nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87	nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88	nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.modules.database_id =</b> dbo.databases.database_id	FK_modules_database

## Linked from

	Table	Join	Title / Name / Description
→	dbo.erd_links (ERD links)	<b>dbo.modules.module_id = dbo.erd_links.module_id</b>	FK_erd_links_modules
→	dbo.erd_nodes (ERD entities)	<b>dbo.modules.module_id = dbo.erd_nodes.module_id</b>	FK_erd_nodes_modules
→	dbo.erd_nodes_columns (ERD entities columns)	<b>dbo.modules.module_id = dbo.erd_nodes_columns.module_id</b>	FK_erd_nodes_columns_modules
→	dbo.procedures_modules (Procedures-modules)	<b>dbo.modules.module_id = dbo.procedures_modules.module_id</b>	FK_procedures_modules_module
→	dbo.tables_modules (Tables-modules)	<b>dbo.modules.module_id = dbo.tables_modules.module_id</b>	FK_tables_modules_module

## Unique keys

	Columns	Name / Description
🔑	module_id	PK_modules Table primary key

## Triggers

	Name	When	Description
⚡	trg_modules_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.4. Table: dbo.parameters (Parameters)

### Function and procedure parameters.

#### Columns

	Name	Data type	Description / Attributes
☰	parameter_id	int	Row id <b>Identity / Auto increment</b>
☰	procedure_id	int	Parameter's procedure <b>References:</b> dbo.procedures
☰	name	nvarchar(250)	Parameter name, "Returns" for functions output
☰	ordinal_position	int	Position of parameter in parameters list
☰	parameter_mode	nvarchar(10)	IN, OUT, INOUT
☰	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> N'DBMS'
☰	status	char(1)	'D' if parameter was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
☰	description	nvarchar(MAX)	Parameter description defined with Dataedo <b>Nullable</b>
☰	datatype	nvarchar(250)	Column data type <b>Nullable</b>
☰	data_length	nvarchar(20)	Data length/precision as text. Depending on data type it can have different formats, e.g. "100", "5,4", "MAX" etc. Null if parameter has no length/precision defined. <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
■	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
■	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
■	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17	nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18	nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19	nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20	nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21	nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22	nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23	nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24	nvarchar(MAX)	Custom field 24 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field25	nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26	nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27	nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28	nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29	nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30	nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31	nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34	nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35	nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates

		Name	Data type	Description / Attributes
■	field41		nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42		nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43		nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44		nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45		nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46		nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47		nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48		nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49		nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50		nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51		nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52		nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53		nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54		nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55		nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56		nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57		nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58		nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59		nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60		nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61		nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62		nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63		nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.procedures (Procedures and Functions)	<b>dbo.parameters</b> .procedure_id = dbo.procedures.procedure_id	FK_parameters_procedures
→	dbo.schema_updates (Imports/Updates)	<b>dbo.parameters</b> .update_id = dbo.schema_updates.update_id	fk_parameters_schema_updates

## Linked from

	Table	Join	Title / Name / Description
←	dbo.parameters_changes (Parameters - Schema changes)	<b>dbo.parameters</b> .parameter_id = dbo.parameters_changes.parameter_id	fk_parameters_changes_parameters

## Unique keys

	Columns	Name / Description
🔑	parameter_id	PK_parameters Table primary key

## Triggers

	Name	When	Description
⚡	trg_parameters_change_track_insert	After Insert	Schema change tracking trigger
⚡	trg_parameters_change_track_update	After Update	Schema change tracking trigger

	Name	When	Description
	trg_parameters_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.5. Table: dbo.procedures (Procedures and Functions)

Database stored procedures and functions.

Procedures and functions are distinguished by *object\_type* column with 'PROCEDURE' and 'FUNCTION' values.

Columns

	Name	Data type	Description / Attributes
	procedure_id	int	Row id <b>Identity / Auto increment</b>
	database_id	int	Documentation which object is part of <b>References:</b> dbo.databases
	schema	nvarchar(250)	DBMS schema name <b>Nullable</b>
	name	nvarchar(250)	Object DBMS name
	object_type	nvarchar(100)	Database object type: PROCEDURE - procedure, FUNCTION - function
	subtype	nvarchar(100)	Procedure/function type. For procedures: PROCEDURE - regular stored procedure PACKAGE - packages in Oracle For functions: FUNCTION - regular user defined function METHOD - methods in Db2 <b>Nullable</b>
	source	nvarchar(50)	Source of the row: DBMS - imported from database, USER - manual defined by user in Dataedo repository <b>Default:</b> N'DBMS'
	status	char(1)	'D' if procedure/function was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
	title	nvarchar(250)	Object title/alias in Dataedo repository <b>Nullable</b>
	description	nvarchar(MAX)	Procedure/function description defined with Dataedo <b>Nullable</b>
	description_plain	nvarchar(MAX)	Holds description in plain text only, without HTML tags <b>Nullable</b>
	description_search	nvarchar(MAX)	Holds description in plain text and lower case. Column used for searching. <b>Nullable</b>
	function_type	nvarchar(100)	Type of function as defined in DBMS, e.g. "Table-valued", "Scalar-valued" <b>Nullable</b>
	language	nvarchar(100)	Function/procedure language code if relevant <b>Nullable</b>
	definition	nvarchar(MAX)	Procedure/function script imported from DBMS <b>Nullable</b>
	dbms_creation_date	datetime	Time procedure/function was created in DBMS <b>Nullable</b>
	dbms_last_modification_date	datetime	Time procedure/function was last updated in DBMS <b>Nullable</b>
	synchronization_date	datetime	Time of last object update/import from DBMS <b>Nullable</b>

	Name	Data type	Description / Attributes
■	synchronized_by	nvarchar(100)	Login of user that last updated/imported the object from DBMS <b>Nullable</b>
■	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
■	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
■	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
■	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17	nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18	nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19	nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20	nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21	nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22	nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23	nvarchar(MAX)	Custom field 23 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field24	nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25	nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26	nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27	nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28	nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29	nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30	nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31	nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34	nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35	nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	exists_in_DBMS	bit	For internal use only <b>Nullable</b>

	Name	Data type	Description / Attributes
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> References: dbo.schema_updates
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57	nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58	nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59	nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60	nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61	nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62	nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63	nvarchar(MAX)	Custom field 63 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field88	nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.procedures.database_id = dbo.databases.database_id</b>	FK_procedures_database
→	dbo.schema_updates (Imports/Updates)	<b>dbo.procedures.update_id = dbo.schema_updates.update_id</b>	fk_procedures_schema_updates

## Linked from

	Table	Join	Title / Name / Description
←	dbo.parameters (Parameters)	<b>dbo.procedures.procedure_id = dbo.parameters.procedure_id</b>	FK_parameters_procedures
←	dbo.parameters_changes (Parameters - Schema changes)	<b>dbo.procedures.procedure_id = dbo.parameters_changes.procedure_id</b>	fk_parameters_changes_procedures
←	dbo.procedures_changes (Procedures and Functions - Schema changes)	<b>dbo.procedures.procedure_id = dbo.procedures_changes.procedure_id</b>	fk_procedures_changes_procedures
←	dbo.procedures_modules (Procedures-modules)	<b>dbo.procedures.procedure_id = dbo.procedures_modules.procedure_id</b>	FK_procedures_modules_procedure

## Unique keys

	Columns	Name / Description
🔑	procedure_id	PK_procedures Table primary key

## Triggers

	Name	When	Description
⚡	trg_procedures_change_track_insert	After Insert	Schema change tracking trigger
⚡	trg_procedures_change_track_update	After Update	Schema change tracking trigger
⚡	trg_procedures_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.6. Table: dbo.procedures\_modules (Procedures-modules)

Many-to-many relationship of procedures and modules.

### Columns

	Name	Data type	Description / Attributes
☰	procedure_module_id	int	Row id <b>Identity / Auto increment</b>
☰	procedure_id	int	Ref: procedures <b>References:</b> dbo.procedures
☰	module_id	int	Ref: modules <b>References:</b> dbo.modules
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
☰	source_id	int	For internal use only <b>Nullable</b>

### Links to

	Table	Join	Title / Name / Description
→	dbo.modules (Modules)	<b>dbo.procedures_modules.module_id</b> = dbo.modules.module_id	FK_procedures_modules_module
→	dbo.procedures (Procedures and Functions)	<b>dbo.procedures_modules.procedure_id</b> d = dbo.procedures.procedure_id	FK_procedures_modules_procedure

### Unique keys

	Columns	Name / Description
🔑	procedure_module_id	PK_procedures_modules Table primary key

	Columns	Name / Description
🔑	procedure_id, module_id	UK_procedures_modules Procedure-module combination must be unique

## Triggers

	Name	When	Description
⚡	trg_procedures_modules_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

### 2.1.1.7. Table: dbo.tables (Tables and Views)

Database tables and views.

Tables and views are distinguished by *object\_type* column with 'TABLE' and 'VIEW' values.

#### Columns

	Name	Data type	Description / Attributes
☰	table_id	int	Row id <b>Identity / Auto increment</b>
☰	database_id	int	Documentation which object is part of. Ref: databases. <b>References:</b> dbo.databases
☰	schema	nvarchar(250)	DBMS schema name <b>Nullable</b>
☰	name	nvarchar(250)	Object name
☰	object_type	nvarchar(100)	Database object type. 'TABLE' for tables, 'VIEW' for views.
☰	subtype	nvarchar(100)	For tables: TABLE - regular table, default EXTERNAL_TABLE - External table FOREIGN_TABLE - Foreign table FILE_TABLE - File table GRAPH_TABLE - Graph table GRAPH_NODE_TABLE - Graph node/vertex table GRAPH_EDGE_TABLE - Graph edge table TEMPORAL_TABLE - Temporal table SYSTEM_VERSIONED_TABLE - System-versioned table HISTORY_TABLE - History table SEARCH_INDEX - Search index (as in Elasticsearch) COLLECTION - Collection (as in MongoDB) ENTITY - Entity CUBE - Cube DIMENSION - Dimension FLAT_FILE - Flat file OBJECT - Object STANDARD_OBJECT - Standard object CUSTOM_OBJECT - Custom object EXTERNAL_OBJECT - External object For views: VIEW - regular view, default MATERIALIZED_VIEW - Materialized view EDITIONING_VIEW - Editioning view INDEXED_VIEW - Indexed view NAMED_QUERY - Named query <b>Nullable</b>
☰	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> N'DBMS'
☰	location	nvarchar(MAX)	Table physical location <b>Nullable</b>

	Name	Data type	Description / Attributes
■	status	char(1)	'D' if table/view was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
■	definition	nvarchar(MAX)	View script imported from DBMS <b>Nullable</b>
■	title	nvarchar(250)	Object title/alias in Dataedo repository <b>Nullable</b>
■	description	nvarchar(MAX)	Table/view description defined with Dataedo <b>Nullable</b>
■	description_plain	nvarchar(MAX)	Holds description in plain text only, without HTML tags <b>Nullable</b>
■	description_search	nvarchar(MAX)	Holds description in plain text and lower case. Column used for searching. <b>Nullable</b>
■	dbms_creation_date	datetime	Time table/view was created in DBMS. <b>Nullable</b>
■	dbms_last_modification_date	datetime	Time table/view was last updated in DBMS. <b>Nullable</b>
■	synchronization_date	datetime	Time of last object update/import from DBMS. <b>Nullable</b>
■	synchronized_by	nvarchar(100)	Login of user that last updated/imported the object from DBMS. <b>Nullable</b>
■	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
■	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
■	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
■	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field15		nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16		nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17		nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18		nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19		nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20		nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21		nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22		nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23		nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24		nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25		nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26		nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27		nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28		nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29		nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30		nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31		nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32		nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33		nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34		nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35		nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36		nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37		nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38		nvarchar(MAX)	Custom field 38 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field39		nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40		nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date		datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by		nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date		datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by		nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id		int	For internal use only <b>Nullable</b>
■	temp_sync_status		bit	For internal use only <b>Default:</b> 0
■	exists_in_DBMS		bit	For internal use only <b>Nullable</b>
■	update_id		int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	field41		nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42		nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43		nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44		nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45		nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46		nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47		nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48		nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49		nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50		nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51		nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52		nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53		nvarchar(MAX)	Custom field 53 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57	nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58	nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59	nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60	nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61	nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62	nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63	nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64	nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65	nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66	nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67	nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68	nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69	nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70	nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71	nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72	nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73	nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74	nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75	nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76	nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77	nvarchar(MAX)	Custom field 77 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89		nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90		nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91		nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92		nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93		nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94		nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95		nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96		nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97		nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98		nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99		nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100		nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.tables.database_id =</b> dbo.databases.database_id	FK_tables_databases
→	dbo.schema_updates (Imports/Updates)	<b>dbo.tables.update_id =</b> dbo.schema_updates.update_id	fk_tables_schema_updates

## Linked from

	Table	Join	Title / Name / Description
←	dbo.columns (Columns)	<b>dbo.tables.table_id =</b> dbo.columns.table_id	FK_column_table
←	dbo.columns_changes (Columns - Schema changes)	<b>dbo.tables.table_id =</b> dbo.columns_changes.table_id	fk_columns_changes_tables
←	dbo.erd_nodes (ERD entities)	<b>dbo.tables.table_id =</b> dbo.erd_nodes.table_id	FK_erd_nodes_tables
←	dbo.glossary_mappings (Term-Data links)	<b>dbo.tables.table_id =</b> dbo.glossary_mappings.object_id	fk_tables_glossary_mappings
←	dbo.tables_changes (Tables and Views - Schema changes)	<b>dbo.tables.table_id =</b> dbo.tables_changes.table_id	fk_tables_changes_tables
←	dbo.tables_modules (Tables-modules)	<b>dbo.tables.table_id =</b> dbo.tables_modules.table_id	FK_tables_modules_tables
←	dbo.tables_relations (Relationships)	<b>dbo.tables.table_id =</b> dbo.tables_relations.fk_table_id	Foreign table FK_tables_relations_table_fk
←	dbo.tables_relations (Relationships)	<b>dbo.tables.table_id =</b> dbo.tables_relations.pk_table_id	Primary table FK_tables_relations_table_pk
←	dbo.tables_relations_changes (Relationships - Schema changes)	<b>dbo.tables.table_id =</b> dbo.tables_relations_changes.pk_table_id	fk_tables_relations_changes_tables
←	dbo.tables_relations_changes (Relationships - Schema changes)	<b>dbo.tables.table_id =</b> dbo.tables_relations_changes.fk_table_id	fk_tables_relations_changes_tables
←	dbo.triggers (Triggers)	<b>dbo.tables.table_id =</b> dbo.triggers.table_id	FK_triggers_table
←	dbo.triggers_changes (Triggers - Schema changes)	<b>dbo.tables.table_id =</b> dbo.triggers_changes.table_id	fk_triggers_changes_tables
←	dbo.unique_constraints (Unique keys)	<b>dbo.tables.table_id =</b> dbo.unique_constraints.table_id	FK_unique_constraints_tables
←	dbo.unique_constraints_changes (Unique keys - Schema changes)	<b>dbo.tables.table_id =</b> dbo.unique_constraints_changes.table_id	fk_unique_constraints_changes_tables

## Unique keys

	Columns	Name / Description
🔑	table_id	PK_tables Table primary key

## Triggers

	Name	When	Description
⚡	trg_tables_change_track_insert	After Insert	Schema change tracking trigger

	Name	When	Description
	trg_tables_change_track_update	After Update	Schema change tracking trigger
	trg_tables_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.8. Table: dbo.tables\_modules (Tables-modules)

Many-to-many relationship of tables and modules.

Columns

	Name	Data type	Description / Attributes
	table_module_id	int	Row id <b>Identity / Auto increment</b>
	table_id	int	Ref: tables <b>References:</b> dbo.tables
	module_id	int	Ref: modules <b>References:</b> dbo.modules
	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
	source_id	int	For internal use only <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→	dbo.modules (Modules)	<b>dbo.tables_modules.module_id = dbo.modules.module_id</b>	FK_tables_modules_module
→	dbo.tables (Tables and Views)	<b>dbo.tables_modules.table_id = dbo.tables.table_id</b>	FK_tables_modules_tables

Unique keys

	Columns	Name / Description
	table_module_id	PK_tables_modules Table primary key
	table_id, module_id	UK_tables_modules Table-module combination must be unique

Triggers

	Name	When	Description
	trg_tables_modules_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

### 2.1.1.9. Table: dbo.tables\_relations (Relationships)

Holds table relationships (foreign keys), those defined in DBMS and those defined by user in Dataedo. The two types are distinguished by source column (DBMS/USER).

#### Columns

		Name	Data type	Description / Attributes
		table_relation_id	int	Row id <b>Identity / Auto increment</b>
		pk_table_id	int	Primary key table <b>References:</b> dbo.tables
		fk_table_id	int	Foreign key table <b>References:</b> dbo.tables
		name	nvarchar(250)	Relation name from DBMS or assigned by user if relation is user defined
		source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> 'USER'
		status	char(1)	'D' if relation was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
		title	nvarchar(250)	Relationship title <b>Nullable</b>
		description	nvarchar(MAX)	Relation description defined with Dataedo. <b>Nullable</b>
		fk_type	nvarchar(15)	Foreign key table cardinality: MANY (default), ONE <b>Default:</b> N'MANY'
		pk_type	nvarchar(15)	Primary key table cardinality: Currently available only ONE <b>Default:</b> N'ONE'
		disabled	bit	Is imported foreign key disabled in DBMS (1 - yes) <b>Nullable</b>
		update_rule	nvarchar(100)	Update rule defined in DBMS: NO ACTION, SET NULL, CASCADE <b>Nullable</b>
		delete_rule	nvarchar(100)	Delete rule defined in DBMS: NO ACTION, SET NULL, CASCADE <b>Nullable</b>
		field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
		field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
		field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
		field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
		field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
		field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
		field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
		field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
		field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17	nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18	nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19	nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20	nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21	nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22	nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23	nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24	nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25	nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26	nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27	nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28	nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29	nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30	nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31	nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field34		nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35		nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36		nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37		nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38		nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39		nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40		nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date		datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by		nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date		datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by		nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id		int	For internal use only <b>Nullable</b>
■	temp_sync_status		bit	For internal use only <b>Default:</b> 0
■	update_id		int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	field41		nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42		nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43		nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44		nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45		nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46		nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47		nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48		nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49		nvarchar(MAX)	Custom field 49 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57	nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58	nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59	nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60	nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61	nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62	nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63	nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64	nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65	nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66	nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67	nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68	nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69	nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70	nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71	nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72	nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73	nvarchar(MAX)	Custom field 73 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89		nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90		nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91		nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92		nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93		nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94		nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95		nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96		nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97		nvarchar(MAX)	Custom field 97 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field98		nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99		nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100		nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→ i	dbo.schema_updates (Imports/Updates)	<b>dbo.tables_relations.update_id</b> = dbo.schema_updates.update_id	fk_tables_relations_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.tables_relations.fk_table_id</b> = dbo.tables.table_id	Foreign table FK_tables_relations_table_fk
→	dbo.tables (Tables and Views)	<b>dbo.tables_relations.pk_table_id</b> = dbo.tables.table_id	Primary table FK_tables_relations_table_pk

## Linked from

	Table	Join	Title / Name / Description
→	dbo.erd_links (ERD links)	<b>dbo.tables_relations.table_relation_id</b> = dbo.erd_links.relation_id	FK_erd_links_tables_relations
→ i	dbo.tables_relations_changes (Relationships - Schema changes)	<b>dbo.tables_relations.table_relation_id</b> = dbo.tables_relations_changes.table_relation_id	fk_tables_relations_changes_tables_relations
→	dbo.tables_relations_columns (Relationships columns)	<b>dbo.tables_relations.table_relation_id</b> = dbo.tables_relations_columns.table_relation_id	FK_tables_relations_columns_relation
→ i	dbo.tables_relations_columns_changes (Relationships columns - Schema changes)	<b>dbo.tables_relations.table_relation_id</b> = dbo.tables_relations_columns_changes.table_relation_id	fk_tables_relations_columns_changes_tables_relations

## Unique keys

	Columns	Name / Description
🔑	table_relation_id	PK_tables_relations Table primary key

## Triggers

	Name	When	Description
⚡	trg_tables_relations_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

### 2.1.1.10. Table: dbo.tables\_relations\_columns (Relationships columns)

Stores foreign keys columns. Each row represents one column pair: column\_fk\_id and column\_pk\_id.

## Columns

		Name	Data type	Description / Attributes
■	🔑	table_relation_column_id	int	Row id <b>Identity / Auto increment</b>

	Name	Data type	Description / Attributes
■	table_relation_id	int	Link to table relation row <b>References:</b> dbo.tables_relations
■	column_fk_id	int	Foreign key table column id <b>References:</b> dbo.columns
■	column_pk_id	int	Primary key table column id <b>References:</b> dbo.columns
■	ordinal_position	int	Column position in relationship
■	status	char(1)	'D' if relation column was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates

#### Links to

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.tables_relations_columns.column_fk_id</b> = dbo.columns.column_id	FK_tables_relations_columns_column_fk
→	dbo.columns (Columns)	<b>dbo.tables_relations_columns.column_pk_id</b> = dbo.columns.column_id	FK_tables_relations_columns_column_pk
→	dbo.schema_updates (Imports/Updates)	<b>dbo.tables_relations_columns.update_id</b> = dbo.schema_updates.update_id	fk_tables_relations_columns_schema_updates
→	dbo.tables_relations (Relationships)	<b>dbo.tables_relations_columns.table_relation_id</b> = dbo.tables_relations.table_relation_id	FK_tables_relations_columns_relation

#### Linked from

	Table	Join	Title / Name / Description
→	dbo.tables_relations_columns_changes (Relationships columns - Schema changes)	<b>dbo.tables_relations_columns.table_relation_column_id</b> = dbo.tables_relations_columns_changes.table_relation_column_id	fk_tables_relations_columns_changes_tables_relations_columns

#### Unique keys

	Columns	Name / Description
🔑	table_relation_column_id	PK_tables_relations_columns Table primary key

## Triggers

	Name	When	Description
	trg_tables_relations_cols_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

### 2.1.1.11. Table: dbo.triggers (Triggers)

Holds table triggers.

#### Columns

	Name	Data type	Description / Attributes
	trigger_id	int	Row id <b>Identity / Auto increment</b>
	table_id	int	Table the trigger is related to. Ref: tables. <b>References:</b> dbo.tables
	name	nvarchar(250)	Trigger name.
	type	nvarchar(100)	Defines type of object classified in Dataedo as trigger (visible in Trigger tab by table object). TRIGGER - "regular" trigger, RULE - PostgreSQL rule. <b>Default:</b> N'TRIGGER'
	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> N'DBMS'
	status	char(1)	'D' if trigger was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
	description	nvarchar(MAX)	Trigger description defined with Dataedo. <b>Nullable</b>
	before	bit	Is trigger fired before operation (1 - yes) <b>Default:</b> 0
	after	bit	Is trigger fired after operation (1 - yes) <b>Default:</b> 0
	instead_of	bit	Is trigger fired instead of operation (1 - yes) <b>Default:</b> 0
	on_insert	bit	Is trigger fired on insert operation (1 - yes) <b>Default:</b> 0
	on_update	bit	Is trigger fired on update operation (1 - yes) <b>Default:</b> 0
	on_delete	bit	Is trigger fired on delete operation (1 - yes) <b>Default:</b> 0
	disabled	bit	Is trigger disabled in DBMS (1 - yes) <b>Default:</b> 0
	definition	nvarchar(MAX)	Trigger script imported from DBMS <b>Nullable</b>
	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
■	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
■	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
■	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17	nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18	nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19	nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20	nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21	nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22	nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23	nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24	nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25	nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26	nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27	nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28	nvarchar(MAX)	Custom field 28 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field29	nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30	nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31	nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34	nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35	nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field45		nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46		nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47		nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48		nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49		nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50		nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51		nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52		nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53		nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54		nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55		nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56		nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57		nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58		nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59		nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60		nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61		nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62		nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63		nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89		nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90		nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91		nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92		nvarchar(MAX)	Custom field 92 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>
■	subtype	nvarchar(100)	Trigger type: TRIGGER (default) - regular trigger, RULE - rule <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→ ↓	dbo.schema_updates (Imports/Updates)	<b>dbo.triggers.update_id =</b> dbo.schema_updates.update_id	fk_triggers_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.triggers.table_id =</b> dbo.tables.table_id	FK_triggers_table

Linked from

	Table	Join	Title / Name / Description
→ ↓	dbo.triggers_changes (Triggers - Schema changes)	<b>dbo.triggers.trigger_id =</b> dbo.triggers_changes.trigger_id	fk_triggers_changes_triggers

Unique keys

	Columns	Name / Description
🔑	trigger_id	PK_triggers Table primary key

Triggers

	Name	When	Description
⚡	trg_triggers_change_track_insert	After Insert	Schema change tracking trigger
⚡	trg_triggers_change_track_update	After Update	Schema change tracking trigger
⚡	trg_triggers_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.12. Table: dbo.unique\_constraints (Unique keys)

Holds table unique keys, either defined in DBMS or by user in Dataedo. The two types are distinguished by source column (DBMS/USER).

## Columns

	Name	Data type	Description / Attributes
💡	unique_constraint_id	int	Row id <b>Identity / Auto increment</b>
💡	table_id	int	Table to which unique key belongs <b>References:</b> dbo.tables
💡	name	nvarchar(250)	Unique constraint name from DBMS or assigned by user if constraint is user defined
💡	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> 'USER'
💡	status	char(1)	'D' if unique key was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
💡	primary_key	bit	1 when key is primary key, 0 when regular unique key
💡	disabled	bit	1 when imported key is disabled (only DBMS) <b>Nullable</b>
💡	description	nvarchar(MAX)	Unique key description defined with Dataedo <b>Nullable</b>
💡	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
💡	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
💡	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
💡	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
💡	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
💡	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
💡	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>
💡	field8	nvarchar(MAX)	Custom field 8 <b>Nullable</b>
💡	field9	nvarchar(MAX)	Custom field 9 <b>Nullable</b>
💡	field10	nvarchar(MAX)	Custom field 10 <b>Nullable</b>
💡	field11	nvarchar(MAX)	Custom field 11 <b>Nullable</b>
💡	field12	nvarchar(MAX)	Custom field 12 <b>Nullable</b>
💡	field13	nvarchar(MAX)	Custom field 13 <b>Nullable</b>
💡	field14	nvarchar(MAX)	Custom field 14 <b>Nullable</b>
💡	field15	nvarchar(MAX)	Custom field 15 <b>Nullable</b>
💡	field16	nvarchar(MAX)	Custom field 16 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field17		nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18		nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19		nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20		nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21		nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22		nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23		nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24		nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25		nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26		nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27		nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28		nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29		nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30		nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31		nvarchar(MAX)	Custom field 31 <b>Nullable</b>
■	field32		nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33		nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34		nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35		nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36		nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37		nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38		nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39		nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40		nvarchar(MAX)	Custom field 40 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>
■	field51	nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52	nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53	nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54	nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55	nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56	nvarchar(MAX)	Custom field 56 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field57	nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58	nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59	nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60	nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61	nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62	nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63	nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64	nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65	nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66	nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67	nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68	nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69	nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70	nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71	nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72	nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73	nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74	nvarchar(MAX)	Custom field 74 <b>Nullable</b>
■	field75	nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76	nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77	nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78	nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79	nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80	nvarchar(MAX)	Custom field 80 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field81	nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82	nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83	nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84	nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85	nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86	nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87	nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88	nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89	nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90	nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91	nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92	nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93	nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94	nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95	nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96	nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97	nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98	nvarchar(MAX)	Custom field 98 <b>Nullable</b>
■	field99	nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100	nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→ ↳	dbo.schema_updates (Imports/Updates)	<b>dbo.unique_constraints.update_id =</b> dbo.schema_updates.update_id	fk_unique_constraints_schema_updates
→ ↳	dbo.tables (Tables and Views)	<b>dbo.unique_constraints.table_id =</b> dbo.tables.table_id	FK_unique_constraints_tables

## Linked from

	Table	Join	Title / Name / Description
↳	dbo.unique_constraints_changes (Unique keys - Schema changes)	<b>dbo.unique_constraints.unique_constraint_id =</b> dbo.unique_constraints_changes.unique_constraint_id	fk_unique_constraints_changes_unique_constraints
↳	dbo.unique_constraints_columns (Unique keys columns)	<b>dbo.unique_constraints.unique_constraint_id =</b> dbo.unique_constraints_columns.unique_constraint_id	FK_unique_constraints_columns_unique_constraints
↳	dbo.unique_constraints_columns_changes (Unique keys columns - Schema changes)	<b>dbo.unique_constraints.unique_constraint_id =</b> dbo.unique_constraints_columns_changes.unique_constraint_id	fk_unique_constraints_columns_changes_unique_constraints

## Unique keys

	Columns	Name / Description
🔑	unique_constraint_id	PK_unique_constraints Table primary key

## Triggers

	Name	When	Description
⚡	trg_unique_constraints_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

### 2.1.1.13. Table: dbo.unique\_constraints\_columns (Unique keys columns)

Stores unique key columns.

## Columns

	Name	Data type	Description / Attributes
🔑	unique_constraint_column_id	int	Row id <b>Identity / Auto increment</b>
🔑	unique_constraint_id	int	Ref: unique key <b>References:</b> dbo.unique_constraints
🔑	column_id	int	Ref: table/view column <b>References:</b> dbo.columns
🔑	ordinal_position	int	Column position in index/constraint
🔑	status	char(1)	'D' if unique key column was deleted from DBMS, 'A' otherwise <b>Default:</b> 'A'
🔑	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
🔑	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
🔑	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
🔑	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
🔑	source_id	int	For internal use only <b>Nullable</b>

	Name	Data type	Description / Attributes
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0
■	update_id	int	Id of the last schema import/update from source database where element was created/modified <b>Nullable</b> <b>References:</b> dbo.schema_updates

Links to

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.unique_constraints_columns.column_id = dbo.columns.column_id</b>	FK_unique_constraints_columns_columns
→ ↳	dbo.schema_updates (Imports/Updates)	<b>dbo.unique_constraints_columns.update_id = dbo.schema_updates.update_id</b>	fk_unique_constraints_columns_schema_updates
→	dbo.unique_constraints (Unique keys)	<b>dbo.unique_constraints.unique_constraint_id = dbo.unique_constraints.unique_constraint_id</b>	FK_unique_constraints_columns_unique_constraints

Linked from

	Table	Join	Title / Name / Description
→ ↳	dbo.unique_constraints_columns_changes (Unique keys columns - Schema changes)	<b>dbo.unique_constraints_columns.unique_constraint_column_id = dbo.unique_constraints_columns_changes.unique_constraint_column_id</b>	fk_unique_constraints_columns_changes_unique_constraints_columns

Unique keys

	Columns	Name / Description	
🔑	unique_constraint_column_id	PK_unique_constraints_columns	Table primary key

Triggers

	Name	When	Description
⚡	trg_unique_constraints_cols_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.1.1.14. Table: dbo.user\_connections (User connection details)

Table holds user connection details assigned to specific documentation/database (databases table).

Columns

	Name	Data type	Description / Attributes
■	connection_id	int	Row id <b>Identity / Auto increment</b>
■	license_id	int	User id. Ref: licenses <b>References:</b> dbo.licenses
■	database_id	int	Related documentation. Ref: databases <b>References:</b> dbo.databases
■	database_name	nvarchar(1024)	Database/schema name <b>Nullable</b>
■	user	nvarchar(1024)	Username used to connect to database <b>Nullable</b>

	Name	Data type	Description / Attributes
■	password	nvarchar(MAX)	Encrypted password, null if not saved <b>Nullable</b>
■	windows_authentication	bit	Used for SQL Server: 1 if Windows authentication, 0 for SQL Server authentication <b>Nullable</b>
■	different_schema	bit	Used for Oracle: 1 if user selected schema different than user, 0 otherwise <b>Nullable</b>
■	connection_type	nvarchar(100)	Used for Oracle: connection type: BASIC for host/port/service name, TNS for network alias <b>Nullable</b>
■	host	nvarchar(1024)	Host name/address <b>Nullable</b>
■	port	int	Port for Oracle connection <b>Nullable</b>
■	service_name	nvarchar(100)	Service name for Oracle connection <b>Nullable</b>
■	network_alias	nvarchar(100)	Network alias for Oracle TNS connection <b>Nullable</b>
■	filter	nvarchar(MAX)	Import filter definiton saved in XML format <b>Nullable</b>
■	multiple_schemas	bit	Used for Oracle: 1 if multiple schemas were selected <b>Nullable</b>
■	source_id	int	For internal use only <b>Nullable</b>
■	ssl_key_path	nvarchar(500)	Path of the file that contains X.509 key <b>Nullable</b>
■	ssl_cert_path	nvarchar(500)	Path of the file that contains X.509 certificate <b>Nullable</b>
■	ssl_ca_path	nvarchar(500)	Path of the file that contains trusted SSL Certificate Authority certificate <b>Nullable</b>
■	ssl_cipher	nvarchar(MAX)	List of permitted ciphers for connection encryption <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	instance_identifier	nvarchar(50)	Identifies which parameters to use for connection: SERVICE_NAME - service_name, SID - oracle_sid <b>Nullable</b>
■	oracle_sid	nvarchar(50)	SID for Oracle connection <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.user_connections.database_id</b> = dbo.databases.database_id	FK_user_connections_databases
→	dbo.licenses (Users)	<b>dbo.user_connections.license_id</b> = dbo.licenses.license_id	FK_user_connections_licenses

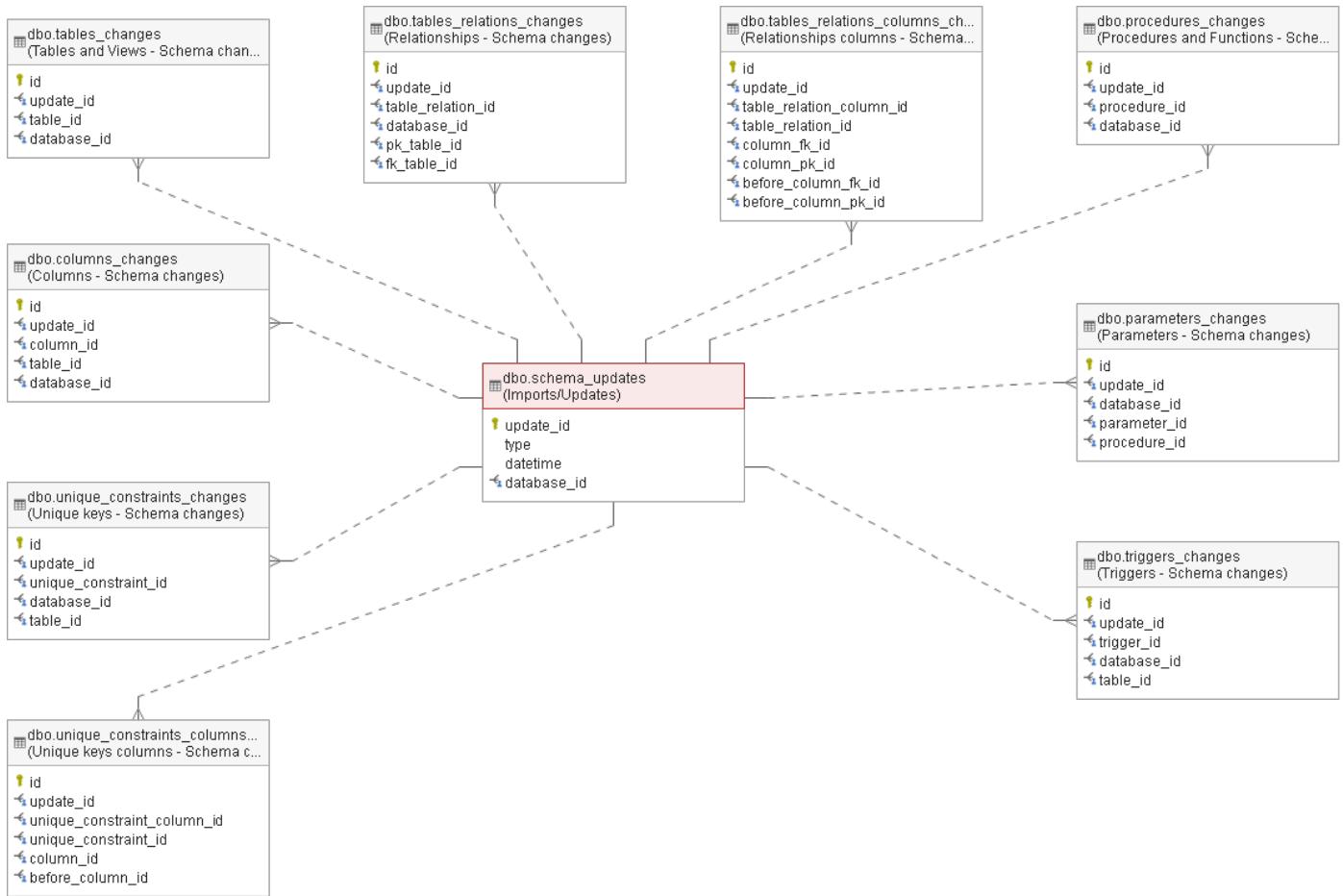
## Unique keys

	Columns	Name / Description
🔑	connection_id	PK_user_connections
🔑	license_id, database_id	UK_user_connections_license_id_database_id

## Triggers

	Name	When	Description
⚡	trg_user_connections_Modify	After Insert, Update	

## 2.2. Schema Change Tracking



Schema Changes Tracking mechanism in Dataedo detects changes to database schema (tables, columns, data types etc) at import time and stores it in history tables.

### History tables

Changes are stored in history tables, one for each schema table. Each table has **\_changes** postfix:

- **tables -> tables\_changes**,
- **columns -> columns\_changes**,
- etc.

### Updates

Each row in history table represents one change detected at schema import operation from source database. Those import operations are stored in the **schema\_updates** table. Each Import/Update operation is logged as one row and the type of the operation is stored in **type** column (IMPORT, UPDATE). All the changes from one import operation can be grouped with **update\_id** column that is also primary key of **schema\_updates** table.

### Operation types

There are three types of change and row types in history tables and this information is stored in **operation** column:

- ADDED - object/element was added,
- UPDATED - object/element was updated,
- DELETED - object/element was deleted

### Change time

There are two columns that you can use to find the time of the changes:

- **valid\_from** - when change was detected in the repository
- **valid\_to** - when next change of this object was detected

Please note that because import time can last significant amount of time times in different rows and tables from the same update may vary. You can also use **update\_id->schema\_updates.datetime**.

## Attribute changes - before/after values

History tables hold details of changed object/element attributes, both after and before change values, like in the following example:

- After change: datatype
- Before change: before\_datatype

## Initial import

Initial database import is not logged in schema change tables.

### 2.2.1. Tables

#### 2.2.1.1. Table: dbo.columns\_changes (Columns - Schema changes)

Schema changes detected in table/view columns detected at import time.

##### Columns

	Name	Data type	Description / Attributes
■	 id	int	Row id <b>Identity / Auto increment</b>
■	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	column_id	int	Updated column id <b>References:</b> dbo.columns
■	table_id	int	Updated column table id <b>References:</b> dbo.tables
■	database_id	int	Updated column documentation id <b>Nullable</b> <b>References:</b> dbo.databases
■	table_schema	nvarchar(250)	Updated column table schema name - after update <b>Nullable</b>
■	table_name	nvarchar(250)	Updated column table name - after update <b>Nullable</b>
■	column_name	nvarchar(250)	Updated column name - after update
■	ordinal_position	int	Updated column ordinal position - after update <b>Nullable</b>
■	datatype	nvarchar(250)	Updated column data type - after update <b>Nullable</b>
■	data_length	nvarchar(50)	Updated column data length - after update <b>Nullable</b>
■	nullable	bit	Updated column nullability - after update
■	default_value	nvarchar(MAX)	Updated column default value - after update <b>Nullable</b>

	Name	Data type	Description / Attributes
■	is_identity	bit	Updated column identity flag - after update
■	is_computed	bit	Updated column computed flag - after update
■	computed_formula	nvarchar(MAX)	Updated column computed formula - after update <b>Nullable</b>
■	before_column_name	nvarchar(250)	Updated column name - before update <b>Nullable</b>
■	before_ordinal_position	int	Updated column ordinal position - before update <b>Nullable</b>
■	before_datatype	nvarchar(250)	Updated column data type - before update <b>Nullable</b>
■	before_data_length	nvarchar(50)	Updated column data length - before update <b>Nullable</b>
■	before_nullable	bit	Updated column nullability - before update <b>Nullable</b>
■	before_default_value	nvarchar(MAX)	Updated column default value - before update <b>Nullable</b>
■	before_is_identity	bit	Updated column identity flag - before update <b>Nullable</b>
■	before_is_computed	bit	Updated column computed flag - before update <b>Nullable</b>
■	before_computed_formula	nvarchar(MAX)	Updated column computed formula - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.columns_changes.column_id = dbo.columns.column_id</b>	fk_columns_changes_columns
→	dbo.databases (Documentations)	<b>dbo.columns_changes.database_id = dbo.databases.database_id</b>	fk_columns_changes_databases
→	dbo.schema_updates (Imports/Updates)	<b>dbo.columns_changes.update_id = dbo.schema_updates.update_id</b>	fk_columns_changes_schema_updates

	Table	Join	Title / Name / Description
→ + 1	dbo.tables (Tables and Views)	dbo.columns_changes.table_id = dbo.tables.table_id	fk_columns_changes_tables

## Unique keys

	Columns	Name / Description
🔑	id	PK_columns_changes

### 2.2.1.2. Table: dbo.parameters\_changes (Parameters - Schema changes)

Schema changes detected in procedures/function paramters detected at import time.

#### Columns

	Name	Data type	Description / Attributes
☰	🔑 id	int	Row id <b>Identity / Auto increment</b>
☰	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
☰	database_id	int	Updated parameter documentation id <b>References:</b> dbo.databases
☰	parameter_id	int	Updated parameter id <b>References:</b> dbo.parameters
☰	procedure_id	int	Updated parameter procedure/function id <b>References:</b> dbo.procedures
☰	ordinal_position	int	Updated parameter ordinal position - after update
☰	parameter_mode	nvarchar(10)	Updated parameter mode IN, OUT, INOUT - after update
☰	name	nvarchar(250)	Updated parameter name - after update
☰	datatype	nvarchar(250)	Updated parameter data type - after update <b>Nullable</b>
☰	data_length	nvarchar(20)	Updated parameter data length - after update <b>Nullable</b>
☰	before_ordinal_position	int	Updated parameter ordinal position - before update <b>Nullable</b>
☰	before_parameter_mode	nvarchar(10)	Updated parameter mode IN, OUT, INOUT - before update <b>Nullable</b>
☰	before_datatype	nvarchar(250)	Updated parameter data type - before update <b>Nullable</b>
☰	before_data_length	nvarchar(20)	Updated parameter data length - before update <b>Nullable</b>
☰	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
☰	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
☰	valid_to	datetime	Timestamp when this change ends it's period of validity and new change of this object was detected <b>Nullable</b>
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()

	Name	Data type	Description / Attributes
[[table]]	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
[[table]]	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
[[table]]	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
[[table]]	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
↳ ↴	dbo.databases (Documentations)	<b>dbo.parameters_changes.database_id</b> = dbo.databases.database_id	fk_parameters_changes_databases
↳ ↴	dbo.parameters (Parameters)	<b>dbo.parameters_changes.parameter_id</b> = dbo.parameters.parameter_id	fk_parameters_changes_parameters
↳ ↴	dbo.procedures (Procedures and Functions)	<b>dbo.parameters_changes.procedure_id</b> = dbo.procedures.procedure_id	fk_parameters_changes_procedures
↳ ↴	dbo.schema_updates (Imports/Updates)	<b>dbo.parameters_changes.update_id</b> = dbo.schema_updates.update_id	fk_parameters_changes_schema_updates

Unique keys

	Columns	Name / Description
🔑	id	PK_parameters_changes

## 2.2.1.3. Table: dbo.procedures\_changes (Procedures and Functions - Schema changes)

Schema changes detected in procedures/function detected at import time.

Columns

	Name	Data type	Description / Attributes
[[table]]	id	int	Row id <b>Identity / Auto increment</b>
[[table]]	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
[[table]]	procedure_id	int	Updated procedure/function id <b>References:</b> dbo.procedures
[[table]]	database_id	int	Updated procedure/function documentation id <b>References:</b> dbo.databases
[[table]]	schema	nvarchar(250)	Updated procedure/function schema <b>Nullable</b>
[[table]]	name	nvarchar(250)	Updated procedure/function name
[[table]]	object_type	nvarchar(100)	Updated procedure/function object type: PROCEDURE/FUNCTION - after update
[[table]]	subtype	nvarchar(100)	Updated procedure/function type - after update <b>Nullable</b>
[[table]]	function_type	nvarchar(100)	Updated function type - after update <b>Nullable</b>

	Name	Data type	Description / Attributes
■	definition	nvarchar(MAX)	Updated procedure/function definition script - after update <b>Nullable</b>
■	before_object_type	nvarchar(100)	Updated procedure/function object type: PROCEDURE/FUNCTION - before update <b>Nullable</b>
■	before_subtype	nvarchar(100)	Updated procedure/function type - before update <b>Nullable</b>
■	before_function_type	nvarchar(100)	Updated function type - before update <b>Nullable</b>
■	before_definition	nvarchar(MAX)	Updated procedure/function definition script - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	dbms_last_modification_date	datetime	Timestamp when procedure/function was last updated in DBMS before import <b>Nullable</b>
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.procedures_changes.database_id</b> = dbo.databases.database_id	fk_procedures_changes_databases
→	dbo.procedures (Procedures and Functions)	<b>dbo.procedures_changes.procedure_id</b> = dbo.procedures.procedure_id	fk_procedures_changes_procedures
→	dbo.schema_updates (Imports/Updates)	<b>dbo.procedures_changes.update_id</b> = dbo.schema_updates.update_id	fk_procedures_changes_schema_updates

Unique keys

	Columns	Name / Description
🔑	id	PK_procedures_changes

## 2.2.1.4. Table: dbo.schema\_updates (Imports/Updates)

Holds all schema imports/updates from source database.

## Columns

	Name	Data type	Description / Attributes
PK	update_id	int	Id of the schema import/update from source database <b>Identity / Auto increment</b>
	type	nvarchar(100)	Type of import: IMPORT - first schema import, UPDATE - subsequent connection
	datetime	datetime	Time of beginnig of import/update operation <b>Nullable</b> <b>Default:</b> getdate()
	repository_login	nvarchar(1024)	Dataedo username that intiated connection <b>Nullable</b> <b>Default:</b> NULL
	database_id	int	Documentation that was imported/updated <b>Nullable</b> <b>References:</b> dbo.databases
	connection_database_type	nvarchar(100)	Connection database type: ORACLE, SQLSERVER, etc. <b>Nullable</b>
	connection_host	nvarchar(1024)	Connection host name/address <b>Nullable</b>
	connection_user	nvarchar(1024)	Connection username <b>Nullable</b>
	connection_port	int	Connection port number <b>Nullable</b>
	connection_service_name	nvarchar(100)	Connection service name for Oracle connection <b>Nullable</b>
	connection_database_name	nvarchar(MAX)	Connection database name <b>Nullable</b>
	connection_dbms_version	nvarchar(500)	Connection DBMS version as returned by DBMS <b>Nullable</b>
	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.schema_updates.database_id =</b> dbo.databases.database_id	fk_schema_updates_databases

## Linked from

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.schema_updates.update_id =</b> dbo.columns.update_id	fk_columns_schema_updates
→	dbo.columns_changes (Columns - Schema changes)	<b>dbo.schema_updates.update_id =</b> dbo.columns_changes.update_id	fk_columns_changes_schema_updates

	Table	Join	Title / Name / Description
→	dbo.parameters (Parameters)	<b>dbo.schema_updates.update_id = dbo.parameters.update_id</b>	fk_parameters_schema_updates
→	dbo.parameters_changes (Parameters - Schema changes)	<b>dbo.schema_updates.update_id = dbo.parameters_changes.update_id</b>	fk_parameters_changes_schema_updates
→	dbo.procedures (Procedures and Functions)	<b>dbo.schema_updates.update_id = dbo.procedures.update_id</b>	fk_procedures_schema_updates
→	dbo.procedures_changes (Procedures and Functions - Schema changes)	<b>dbo.schema_updates.update_id = dbo.procedures_changes.update_id</b>	fk_procedures_changes_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.schema_updates.update_id = dbo.tables.update_id</b>	fk_tables_schema_updates
→	dbo.tables_changes (Tables and Views - Schema changes)	<b>dbo.schema_updates.update_id = dbo.tables_changes.update_id</b>	fk_tables_changes_schema_updates
→	dbo.tables_relations (Relationships)	<b>dbo.schema_updates.update_id = dbo.tables_relations.update_id</b>	fk_tables_relations_schema_updates
→	dbo.tables_relations_changes (Relationships - Schema changes)	<b>dbo.schema_updates.update_id = dbo.tables_relations_changes.update_id</b>	fk_tables_relations_changes_schema_updates
→	dbo.tables_relations_columns (Relationships columns)	<b>dbo.schema_updates.update_id = dbo.tables_relations_columns.update_id</b>	fk_tables_relations_columns_schema_updates
→	dbo.tables_relations_columns_changes (Relationships columns - Schema changes)	<b>dbo.schema_updates.update_id = dbo.tables_relations_columns_changes.update_id</b>	fk_tables_relations_columns_changes_schema_updates
→	dbo.triggers (Triggers)	<b>dbo.schema_updates.update_id = dbo.triggers.update_id</b>	fk_triggers_schema_updates
→	dbo.triggers_changes (Triggers - Schema changes)	<b>dbo.schema_updates.update_id = dbo.triggers_changes.update_id</b>	fk_triggers_changes_schema_updates
→	dbo.unique_constraints (Unique keys)	<b>dbo.schema_updates.update_id = dbo.unique_constraints.update_id</b>	fk_unique_constraints_schema_updates
→	dbo.unique_constraints_changes (Unique keys - Schema changes)	<b>dbo.schema_updates.update_id = dbo.unique_constraints_changes.update_id</b>	fk_unique_constraints_changes_schema_updates
→	dbo.unique_constraints_columns (Unique keys columns)	<b>dbo.schema_updates.update_id = dbo.unique_constraints_columns.update_id</b>	fk_unique_constraints_columns_schema_updates
→	dbo.unique_constraints_columns_changes (Unique keys columns - Schema changes)	<b>dbo.schema_updates.update_id = dbo.unique_constraints_columns_changes.update_id</b>	fk_unique_constraints_columns_changes_schema_updates

## Unique keys

	Columns	Name / Description
🔑	update_id	PK_schema_updates

### 2.2.1.5. Table: dbo.tables\_changes (Tables and Views - Schema changes)

Schema changes detected in tables/views detected at import time.

#### Columns

		Name	Data type	Description / Attributes
>ID	🔑 id		int	Row id Identity / Auto increment

	Name	Data type	Description / Attributes
■	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	table_id	int	Updated table/view id <b>References:</b> dbo.tables
■	database_id	int	Updated table/view documentation id <b>References:</b> dbo.databases
■	schema	nvarchar(250)	Updated table/view schema - after update <b>Nullable</b>
■	name	nvarchar(250)	Updated table/view name - after update
■	object_type	nvarchar(100)	Updated table/view object type: TABLE/VIEW - after update
■	subtype	nvarchar(100)	Updated table/view type - after update <b>Nullable</b>
■	definition	nvarchar(MAX)	Updated view definition script - after update <b>Nullable</b>
■	before_schema	nvarchar(250)	Updated table/view schema - before update <b>Nullable</b>
■	before_name	nvarchar(250)	Updated table/view name - before update <b>Nullable</b>
■	before_object_type	nvarchar(100)	Updated table/view object type: TABLE/VIEW - before update <b>Nullable</b>
■	before_subtype	nvarchar(100)	Updated table/view type - before update <b>Nullable</b>
■	before_definition	nvarchar(MAX)	Updated view definition script - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	dbms_last_modification_date	datetime	Timestamp when table/view was last updated in DBMS before import <b>Nullable</b>
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	dbo.tables_changes.database_id = dbo.databases.database_id	fk_tables_changes_databases

	Table	Join	Title / Name / Description
→ i	dbo.schema_updates (Imports/Updates)	<b>dbo.tables_changes.update_id =</b> dbo.schema_updates.update_id	fk_tables_changes_schema_updates
→ i	dbo.tables (Tables and Views)	<b>dbo.tables_changes.table_id =</b> dbo.tables.table_id	fk_tables_changes_tables

## Unique keys

	Columns	Name / Description
🔑	id	PK_tables_changes

### 2.2.1.6. Table: dbo.tables\_relations\_changes (Relationships - Schema changes)

Schema changes detected in table relationships detected at import time.

#### Columns

	Name	Data type	Description / Attributes
☰	🔑 id	int	Row id <b>Identity / Auto increment</b>
☰	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
☰	table_relation_id	int	Updated relationship id <b>References:</b> dbo.tables_relations
☰	database_id	int	Updated relationship documentation id <b>References:</b> dbo.databases
☰	pk_table_id	int	Updated relationship primary key table id <b>References:</b> dbo.tables
☰	fk_table_id	int	Updated relationship foreign key table id <b>References:</b> dbo.tables
☰	pk_table_schema	nvarchar(250)	Updated relationship primary key table schema - after update <b>Nullable</b>
☰	fk_table_schema	nvarchar(250)	Updated relationship foreign key table schema - after update <b>Nullable</b>
☰	pk_table_name	nvarchar(250)	Updated relationship primary key table name - after update <b>Nullable</b>
☰	fk_table_name	nvarchar(250)	Updated relationship foreign key table name - after update <b>Nullable</b>
☰	name	nvarchar(250)	Updated relationship name - after update
☰	update_rule	nvarchar(100)	Updated relationship/foreign key update rule defined in DBMS - after update <b>Nullable</b>
☰	delete_rule	nvarchar(100)	Updated relationship/foreign key delete rule defined in DBMS - after update <b>Nullable</b>
☰	disabled	bit	Is updated relationship disabled in DBMS (1 - yes) - after update <b>Nullable</b>
☰	fk_type	nvarchar(15)	Updated relationship foreign key table cardinality - after update
☰	pk_type	nvarchar(15)	Updated relationship primary key table cardinality - after update
☰	before_pk_table_schema	nvarchar(250)	Updated relationship primary key table schema - before update <b>Nullable</b>

	Name	Data type	Description / Attributes
■	before_fk_table_schema	nvarchar(250)	Updated relationship foreign key table schema - before update <b>Nullable</b>
■	before_pk_table_name	nvarchar(250)	Updated relationship primary key table name - before update <b>Nullable</b>
■	before_fk_table_name	nvarchar(250)	Updated relationship foreign key table name - before update <b>Nullable</b>
■	before_name	nvarchar(250)	Updated relationship name - before update <b>Nullable</b>
■	before_update_rule	nvarchar(100)	Updated relationship/foreign key update rule defined in DBMS - before update <b>Nullable</b>
■	before_delete_rule	nvarchar(100)	Updated relationship/foreign key delete rule defined in DBMS - before update <b>Nullable</b>
■	before_disabled	bit	Is updated relationship disabled in DBMS (1 - yes) - before update <b>Nullable</b>
■	before_fk_type	nvarchar(15)	Updated relationship foreign key table cardinality - before update <b>Nullable</b>
■	before_pk_type	nvarchar(15)	Updated relationship primary key table cardinality - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends it's period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.tables_relations_changes.database_id</b> = dbo.databases.database_id	fk_tables_relations_changes_databases
→	dbo.schema_updates (Imports/Updates)	<b>dbo.tables_relations_changes.update_id</b> = dbo.schema_updates.update_id	fk_tables_relations_changes_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.tables_relations_changes.pk_table_id</b> = dbo.tables.table_id	fk_tables_relations_changes_tables
→	dbo.tables (Tables and Views)	<b>dbo.tables_relations_changes.fk_table_id</b> = dbo.tables.table_id	fk_tables_relations_changes_tables

	Table	Join	Title / Name / Description
→ ↳	dbo.tables_relations (Relationships)	dbo.tables_relations_changes.table_relation_id = dbo.tables_relations.table_relation_id	fk_tables_relations_changes_tables_relations

## Unique keys

	Columns	Name / Description
🔑	id	PK_tables_relations_changes

## 2.2.1.7. Table: dbo.tables\_relations\_columns\_changes (Relationships columns - Schema changes)

Schema changes detected in table relationships columns detected at import time.

### Columns

	Name	Data type	Description / Attributes
☰	🔑 id	int	Row id <b>Identity / Auto increment</b>
☰	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
☰	table_relation_column_id	int	Updated relationship column id <b>References:</b> dbo.tables_relations_columns
☰	table_relation_id	int	Updated relationship id <b>References:</b> dbo.tables_relations
☰	column_fk_id	int	Updated relationship foreign key column id - after update <b>References:</b> dbo.columns
☰	column_pk_id	int	Updated relationship primary key column id - after update <b>References:</b> dbo.columns
☰	column_fk_name	nvarchar(250)	Updated relationship foreign key column name - after update <b>Nullable</b>
☰	column_pk_name	nvarchar(250)	Updated relationship primary key column name - after update <b>Nullable</b>
☰	ordinal_position	int	Updated relationship column ordinal position - after update
☰	before_column_fk_id	int	Updated relationship foreign key column id - before update <b>Nullable</b> <b>References:</b> dbo.columns
☰	before_column_pk_id	int	Updated relationship primary key column id - before update <b>Nullable</b> <b>References:</b> dbo.columns
☰	before_column_fk_name	nvarchar(250)	Updated relationship foreign key column name - before update <b>Nullable</b>
☰	before_column_pk_name	nvarchar(250)	Updated relationship primary key column name - before update <b>Nullable</b>
☰	before_ordinal_position	int	Updated relationship column ordinal position - before update <b>Nullable</b>
☰	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
☰	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
☰	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>

	Name	Data type	Description / Attributes
[[table]]	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
[[table]]	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL

Links to

	Table	Join	Title / Name / Description
↳ ↴	dbo.columns (Columns)	<b>dbo.tables_relations_columns_changes</b> .column_fk_id = dbo.columns.column_id	fk_tables_relations_columns_changes_columns
↳ ↴	dbo.columns (Columns)	<b>dbo.tables_relations_columns_changes</b> .column_pk_id = dbo.columns.column_id	fk_tables_relations_columns_changes_columns
↳ ↴	dbo.columns (Columns)	<b>dbo.tables_relations_columns_changes</b> .before_column_fk_id = dbo.columns.column_id	fk_tables_relations_columns_changes_columns
↳ ↴	dbo.columns (Columns)	<b>dbo.tables_relations_columns_changes</b> .before_column_pk_id = dbo.columns.column_id	fk_tables_relations_columns_changes_columns
↳ ↴	dbo.schema_updates (Imports/Updates)	<b>dbo.tables_relations_columns_changes</b> .update_id = dbo.schema_updates.update_id	fk_tables_relations_columns_changes_schema_updates
↳ ↴	dbo.tables_relations (Relationships)	<b>dbo.tables_relations_columns_changes</b> .table_relation_id = dbo.tables_relations.table_relation_id	fk_tables_relations_columns_changes_tables_relations
↳ ↴	dbo.tables_relations_columns (Relationships columns)	<b>dbo.tables_relations_columns_changes</b> .table_relation_column_id = dbo.tables_relations_columns.table_relation_column_id	fk_tables_relations_columns_changes_tables_relations_columns

Unique keys

	Columns	Name / Description
🔑	id	PK_tables_relations_columns_changes

## 2.2.1.8. Table: dbo.triggers\_changes (Triggers - Schema changes)

Schema changes detected in triggers detected at import time.

Columns

	Name	Data type	Description / Attributes
[[table]]	id	int	Row id <b>Identity / Auto increment</b>
[[table]]	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
[[table]]	trigger_id	int	Updated trigger id <b>References:</b> dbo.triggers
[[table]]	database_id	int	Updated trigger documentation id <b>References:</b> dbo.databases
[[table]]	table_id	int	Updated trigger table id <b>References:</b> dbo.tables

	Name	Data type	Description / Attributes
■	name	nvarchar(250)	Updated trigger name
■	type	nvarchar(100)	Updated trigger type
■	before	bit	Is updated trigger fired before operation (1 - yes) - after update
■	after	bit	Is updated trigger fired after operation (1 - yes) - after update
■	instead_of	bit	Is updated trigger fired instead of operation (1 - yes) - after update
■	on_insert	bit	Is updated trigger fired on insert operation (1 - yes) - after update
■	on_update	bit	Is updated trigger fired on update operation (1 - yes) - after update
■	on_delete	bit	Is updated trigger fired on delete operation (1 - yes) - after update
■	disabled	bit	Is updated trigger disabled in DBMS (1 - yes) - after update
■	definition	nvarchar(MAX)	Updated trigger definition script - after update <b>Nullable</b>
■	before_before	bit	Is updated trigger fired before operation (1 - yes) - before update <b>Nullable</b>
■	before_after	bit	Is updated trigger fired after operation (1 - yes) - before update <b>Nullable</b>
■	before_instead_of	bit	Is updated trigger fired instead of operation (1 - yes) - before update <b>Nullable</b>
■	before_on_insert	bit	Is updated trigger fired on insert operation (1 - yes) - before update <b>Nullable</b>
■	before_on_update	bit	Is updated trigger fired on update operation (1 - yes) - before update <b>Nullable</b>
■	before_on_delete	bit	Is updated trigger fired on delete operation (1 - yes) - before update <b>Nullable</b>
■	before_disabled	bit	Is updated trigger disabled (1 - yes) - before update <b>Nullable</b>
■	before_definition	nvarchar(MAX)	Updated trigger definition script - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.triggers_changes.database_id</b> = dbo.databases.database_id	fk_triggers_changes_databases
→	dbo.schema_updates (Imports/Updates)	<b>dbo.triggers_changes.update_id</b> = dbo.schema_updates.update_id	fk_triggers_changes_schema_updates
→	dbo.tables (Tables and Views)	<b>dbo.triggers_changes.table_id</b> = dbo.tables.table_id	fk_triggers_changes_tables
→	dbo.triggers (Triggers)	<b>dbo.triggers_changes.trigger_id</b> = dbo.triggers.trigger_id	fk_triggers_changes_triggers

## Unique keys

	Columns	Name / Description
🔑	id	PK_triggers_changes2

## 2.2.1.9. Table: dbo.unique\_constraints\_changes (Unique keys - Schema changes)

Schema changes detected in table primary/unique keys detected at import time.

### Columns

	Name	Data type	Description / Attributes
>ID	id	int	Row id <b>Identity / Auto increment</b>
ID	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
ID	unique_constraint_id	int	Updated primary/unique key id <b>References:</b> dbo.unique_constraints
ID	database_id	int	Updated primary/unique key documentation id <b>References:</b> dbo.databases
ID	table_id	int	Updated primary/unique key table id <b>References:</b> dbo.tables
ID	name	nvarchar(250)	Updated primary/unique key table name - after update
ID	primary_key	bit	Updated key type flag (1 - primary key, - unique key) - after update
ID	disabled	bit	Is updated key disabled in DBMS (1 - yes) - after update <b>Nullable</b>
ID	before_name	nvarchar(250)	Updated primary/unique key table name - before update <b>Nullable</b>
ID	before_primary_key	bit	Updated key type flag (1 - primary key, - unique key) - before update <b>Nullable</b>
ID	before_disabled	bit	Is updated key disabled in DBMS (1 - yes) - before update <b>Nullable</b>
ID	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
ID	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
ID	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>

	Name	Data type	Description / Attributes
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	description	nvarchar(MAX)	Change user comments <b>Nullable</b>
■	description_date	datetime	Date and time comment field was entered/last updated <b>Nullable</b>
■	description_by	nvarchar(1024)	Username that entered/last updated comment field <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→ ↓	dbo.databases (Documentations)	<b>dbo.unique_constraints_changes.</b> data base_id = dbo.databases.database_id	fk_unique_constraints_changes_databases
→ ↓	dbo.schema_updates (Imports/Updates)	<b>dbo.unique_constraints_changes.</b> upda te_id = dbo.schema_updates.update_id	fk_unique_constraints_changes_schema_updates
→ ↓	dbo.tables (Tables and Views)	<b>dbo.unique_constraints_changes.</b> table _id = dbo.tables.table_id	fk_unique_constraints_changes_tables
→ ↓	dbo.unique_constraints (Unique keys)	<b>dbo.unique_constraints_changes.</b> uni que_constraint_id = dbo.unique_constraints.unique_constr aint_id	fk_unique_constraints_changes_unique_constraints

Unique keys

	Columns	Name / Description
🔑	id	PK_unique_constraints_changes

2.2.1.10. Table: dbo.unique\_constraints\_columns\_changes (Unique keys columns - Schema changes)

Schema changes detected in table primary/unique keys columns detected at import time.

Columns

	Name	Data type	Description / Attributes
■	🔑 id	int	Row id <b>Identity / Auto increment</b>
■	update_id	int	Id of the schema import/update from source database <b>Nullable</b> <b>References:</b> dbo.schema_updates
■	unique_constraint_column_id	int	Updated primary/unique key-column id <b>References:</b> dbo.unique_constraints_columns
■	unique_constraint_id	int	Updated primary/unique key id <b>References:</b> dbo.unique_constraints
■	column_id	int	Updated primary/unique key column id - after update <b>References:</b> dbo.columns
■	column_name	nvarchar(250)	Updated primary/unique key column name - after update <b>Nullable</b>
■	ordinal_position	int	Updated primary/unique key column ordinal position - after update

	Name	Data type	Description / Attributes
■	before_column_id	int	Updated primary/unique key column id - before update <b>Nullable</b> References: dbo.columns
■	before_column_name	nvarchar(250)	Updated primary/unique key column name - before update <b>Nullable</b>
■	before_ordinal_position	int	Updated primary/unique key column ordinal position - before update <b>Nullable</b>
■	operation	nvarchar(50)	Update type: ADDED - object was added, UPDATED - object was updated, DELETED - object was deleted
■	valid_from	datetime	Timestamp when this change was detected <b>Nullable</b>
■	valid_to	datetime	Timestamp when this change ends its period of validity and new change of this object was detected <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL

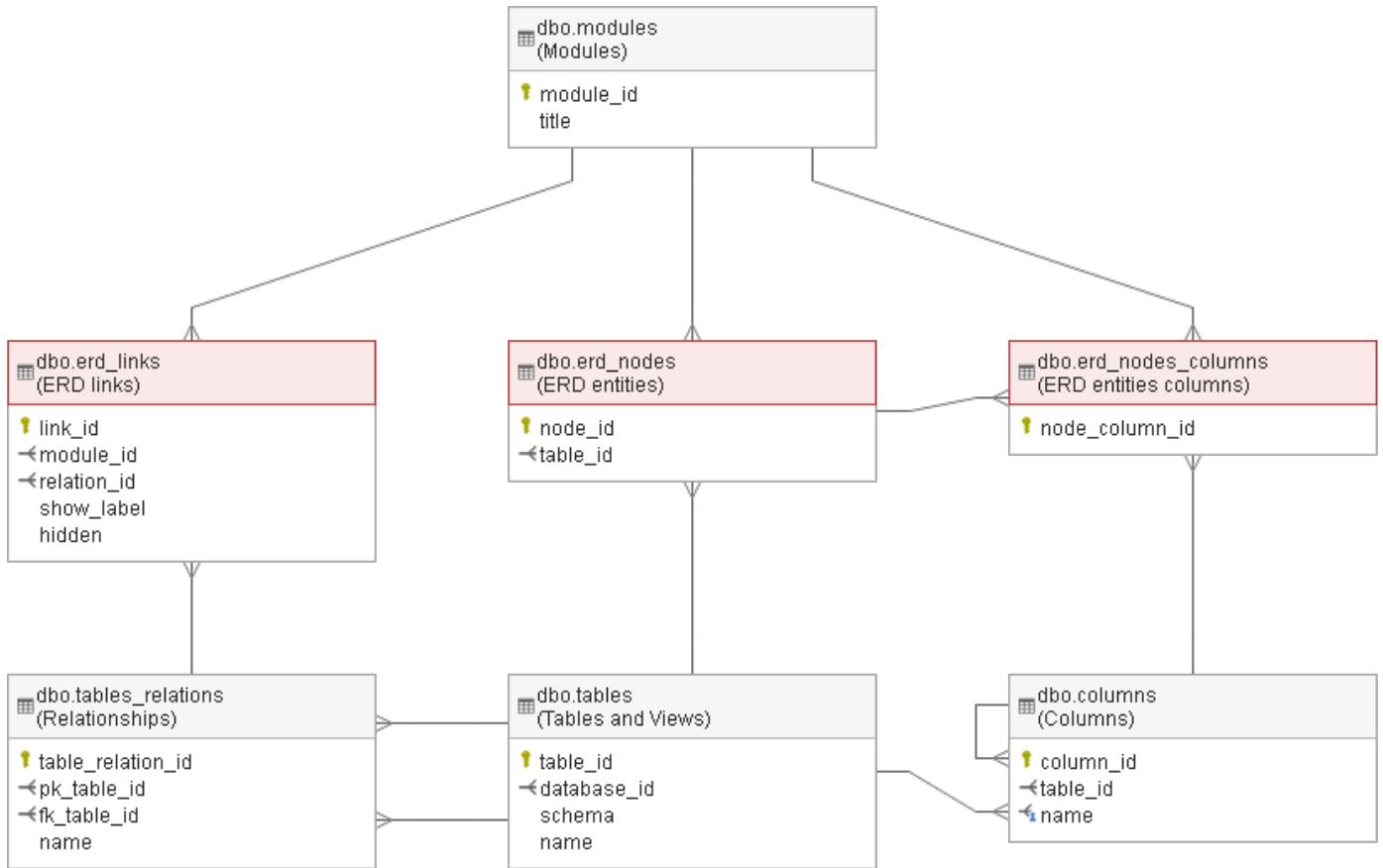
Links to

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.unique_constraints_columns_change</b> <b>ges.column_id =</b> dbo.columns.column_id	fk_unique_constraints_columns_changes_columns
→	dbo.columns (Columns)	<b>dbo.unique_constraints_columns_change</b> <b>ges.before_column_id =</b> dbo.columns.column_id	fk_unique_constraints_columns_changes_columns
→	dbo.schema_updates (Imports/Updates)	<b>dbo.unique_constraints_columns_change</b> <b>ges.update_id =</b> dbo.schema_updates.update_id	fk_unique_constraints_columns_changes_schema_updates
→	dbo.unique_constraints (Unique keys)	<b>dbo.unique_constraints_columns_change</b> <b>ges.unique_constraint_id =</b> dbo.unique_constraints.unique_constraint_id	fk_unique_constraints_columns_changes_unique_constraints
→	dbo.unique_constraints_columns (Unique keys columns)	<b>dbo.unique_constraints_columns_change</b> <b>ges.unique_constraint_column_id =</b> dbo.unique_constraints_columns.unique_constraint_column_id	fk_unique_constraints_columns_changes_unique_constraints_columns

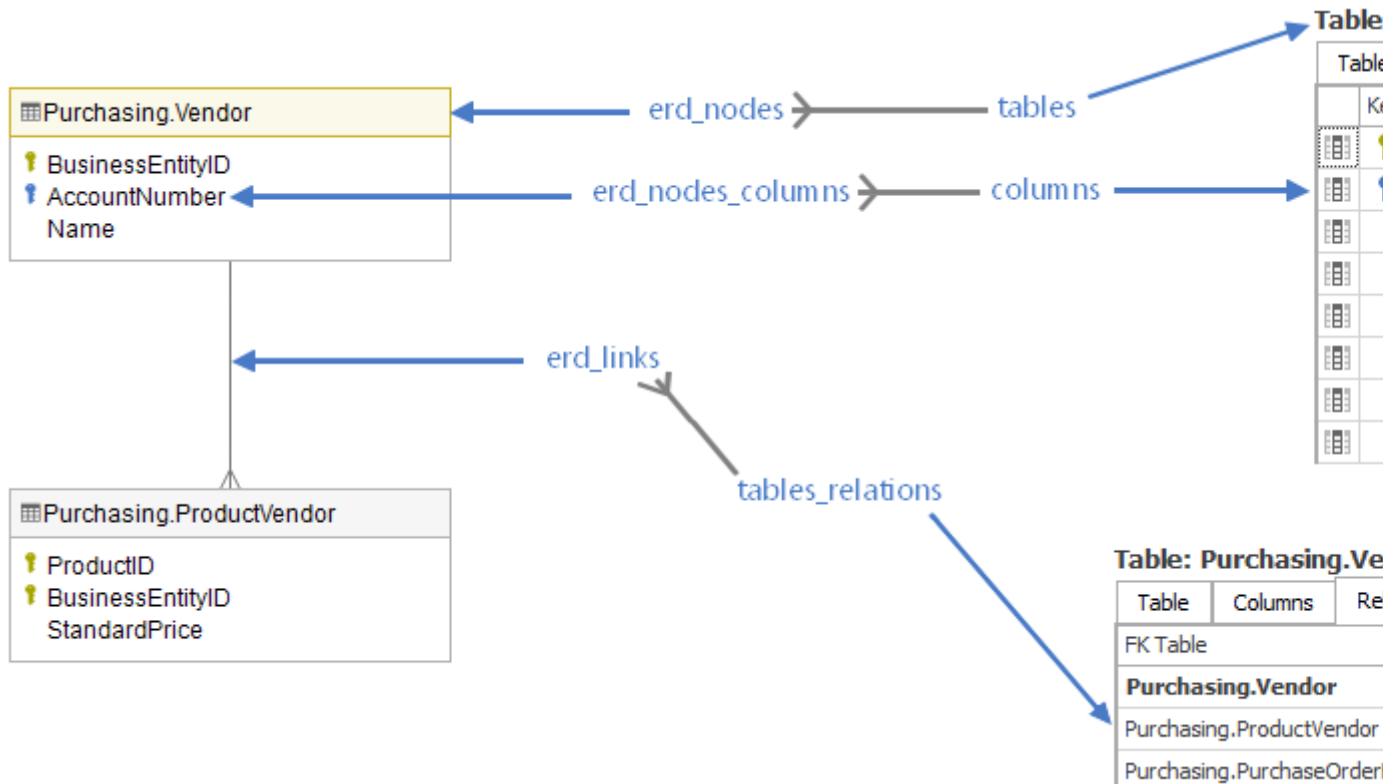
Unique keys

	Columns	Name / Description
🔑	id	PK_unique_constraints_columns_changes

## 2.3. ERDs



Tables describing ERD objects, their position and links.



## 2.3.1. Tables

### 2.3.1.1. Table: dbo.erd\_links (ERD links)

Represents relationships between entities in a specific ERD. Every ERD relationship references a relation defined between tables/views in a Dataedo repository.

This relation is either an imported foreign key defined in source database or a user-defined one in Dataedo repository through ERD or Relations tab.

Each relationship is linked to table/view relations in *tables\_relations* table with *relation\_id* column. ERD is identified by *module\_id* column.

Holds information about link style, link visibility and label visibility.

#### Columns

	Name	Data type	Description / Attributes
link_id	link_id	int	Row id <b>Identity / Auto increment</b>
module_id	module_id	int	Identifier of module/diagram <b>References:</b> dbo.modules
relation_id	relation_id	int	Reference to relation that is represented on diagram <b>References:</b> dbo.tables_relations
show_label	show_label	bit	If 1 then relationship title is displayed on ERD. Default value is 0 <b>Default:</b> 0
show_join_condition	show_join_condition	bit	If 1 then shows relationship join condition on ERD. Default value is 0 <b>Default:</b> 0
link_style	link_style	nvarchar(25)	'STRAIGHT' for a straight link style and 'ORTHOGONAL' for orthogonal one. The former is default. <b>Default:</b> 'STRAIGHT'
hidden	hidden	bit	If 1 then relationship is hidden from ERD. Default value is 0 <b>Default:</b> 0
creation_date	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
created_by	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
last_modification_date	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
modified_by	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
source_id	source_id	int	For internal use only <b>Nullable</b>
label_pos_x	label_pos_x	int	Not used <b>Nullable</b>
label_pos_y	label_pos_y	int	Not used <b>Nullable</b>

#### Links to

	Table	Join	Title / Name / Description
dbo.modules (Modules)	dbo.erd_links.module_id = dbo.modules.module_id		FK_erd_links_modules

	Table	Join	Title / Name / Description
→	dbo.tables_relations (Relationships)	dbo.erd_links.relation_id = dbo.tables_relations.table_relation_id	FK_erd_links_tables_relations

## Unique keys

	Columns	Name / Description
🔑	link_id	PK_erd_links Table primary key
🔑	module_id, relation_id	UK_erd_links_relation Each relation can be added to the diagram only once

## Triggers

	Name	When	Description
⚡	trg_erd_links_descriptions_Modify	After Insert, Update	

### 2.3.1.2. Table: dbo.erd\_nodes (ERD entities)

Represents entity (table or view) in a specific ERD. Each node is linked to a table/view in *tables* table with *table\_id* column.

ERD is identified by *module\_id* column.

Holds information about node position and color.

## Columns

	Name	Data type	Description / Attributes
☰	🔑 node_id	int	Row id <b>Identity / Auto increment</b>
☰	🔑 module_id	int	Identifier of ERD. Ref: modules table <b>References:</b> dbo.modules
☰	🔑 table_id	int	Reference to table/view that is represented on diagram. Ref: tables table <b>References:</b> dbo.tables
☰	pos_x	int	X position on diagram. 0 is the center of a diagram. Can be negative <b>Nullable</b>
☰	pos_y	int	Y position on diagram. 0 is the center of a diagram. Can be negative <b>Nullable</b>
☰	color	char(7)	Box color, e.g. #4CAF50 <b>Nullable</b>
☰	width	int	Box width <b>Nullable</b>
☰	height	int	Box height. Currently not used <b>Nullable</b>
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL

	Name	Data type	Description / Attributes
grid icon	source_id	int	For internal use only <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
→	dbo.modules (Modules)	<b>dbo.erd_nodes.module_id =</b> dbo.modules.module_id	FK_erd_nodes_modules
→	dbo.tables (Tables and Views)	<b>dbo.erd_nodes.table_id =</b> dbo.tables.table_id	FK_erd_nodes_tables

Linked from

	Table	Join	Title / Name / Description
←	dbo.erd_nodes_columns (ERD entities columns)	<b>dbo.erd_nodes.node_id =</b> dbo.erd_nodes_columns.node_id	FK_erd_nodes_columns_erd_nodes

Unique keys

	Columns	Name / Description
key icon	node_id	PK_erd_nodes Table primary key
key icon	module_id, table_id	UK_erd_nodes_table Each table can be added to the diagram only once

Triggers

	Name	When	Description
flash icon	trg_erd_nodes_descriptions_Modify	After Insert, Update	

### 2.3.1.3. Table: dbo.erd\_nodes\_columns (ERD entities columns)

Holds information about visible table/view columns on an ERD - if a column is in this table then it is visible on a diagram.

ERD is identified by *module\_id* column. *Column\_id* references *columns* table.

Columns

	Name	Data type	Description / Attributes
grid icon	node_column_id	int	Row id <b>Identity / Auto increment</b>
grid icon	module_id	int	Identifier of module/diagram <b>References:</b> dbo.modules
grid icon	node_id	int	Ref: erd_nodes table <b>References:</b> dbo.erd_nodes
grid icon	column_id	int	Ref: columns table <b>References:</b> dbo.columns
grid icon	source_id	int	For internal use only <b>Nullable</b>

Links to

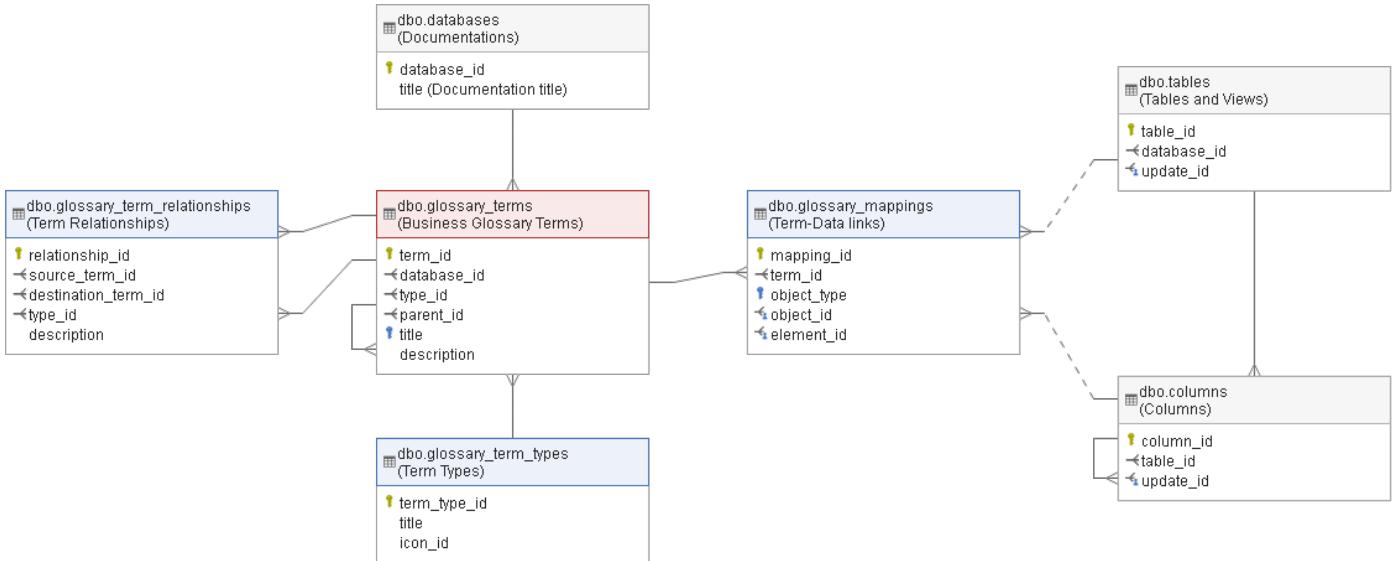
	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.erd_nodes_columns.column_id =</b> dbo.columns.column_id	FK_erd_nodes_columns_columns

	Table	Join	Title / Name / Description
→	dbo.erd_nodes (ERD entities)	<b>dbo.erd_nodes_columns.node_id =</b> dbo.erd_nodes.node_id	FK_erd_nodes_columns_erd_nodes
→	dbo.modules (Modules)	<b>dbo.erd_nodes_columns.module_id =</b> dbo.modules.module_id	FK_erd_nodes_columns_modules

## Unique keys

	Columns	Name / Description
🔑	node_column_id	PK_erd_nodes_columns Table primary key

## 2.4. Business Glossary



Business Glossary tables.

### 2.4.1. Tables

#### 2.4.1.1. Table: dbo.glossary\_mappings (Term-Data links)

Links from **Business Glossary** (terms, policies) to **Data Dictionary** (tables, columns).

##### Columns

		Name	Data type	Description / Attributes
1	🔑	mapping_id	int	Row id <b>Identity / Auto increment</b>
2	🔑	term_id	int	Linked term <b>References:</b> dbo.glossary_terms
3	🔑	object_type	nvarchar(100)	Type of data dictionary element linked: 'COLUMN' - column use element_id to join to columns, 'TABLE' - table, use object_id to join to tables. <b>Nullable</b>
4	🔑	object_id	int	Linked table (tables table) regardless of type of linked object (table or column) <b>References:</b> dbo.tables
5	🔑	element_id	int	Linked column (columns table) if object_type = 'COLUMN', otherwise null <b>Nullable</b> <b>References:</b> dbo.columns
6		creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
7		created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
8		last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
9		modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
10		source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.columns (Columns)	<b>dbo.glossary_mappings.element_id = dbo.columns.column_id</b>	fk_columns_glossary_mappings
→	dbo.glossary_terms (Business Glossary Terms)	<b>dbo.glossary_mappings.term_id = dbo.glossary_terms.term_id</b>	FK_glossary_mappings_glossary_terms
→	dbo.tables (Tables and Views)	<b>dbo.glossary_mappings.object_id = dbo.tables.table_id</b>	fk_tables_glossary_mappings

## Unique keys

	Columns	Name / Description
🔑	mapping_id	PK_glossary_mappings
🔑	term_id, object_type, object_id, element_id	UK_glossary_mappings

## Triggers

	Name	When	Description
⚡	trg_glossary_mappings_Modify	After Insert, Update	

## 2.4.1.2. Table: dbo.glossary\_term\_relationship\_types (Term Relationship Types)

Holds definition of term relationship types.

### Symmetrical relationships

A relationship can be symmetrical (is\_symmetrical = 1), e.g. Related term or bidirectional (is\_symmetrical = 0) e.g. Is a type of/Has a type.

### Relationship name

When showing **source\_term -> destination\_term** use **title** column for relationship name, if the other way around, use **title\_reverse**.

## Columns

	Name	Data type	Description / Attributes
☰	🔑 type_id	int	Row id <b>Identity / Auto increment</b>
☰	title	nvarchar(250)	Relationship name when joining from source_term to destination_term.
☰	title_reverse	nvarchar(250)	Relationship name when joining from destination_term to source_term.
☰	is_symmetrical	bit	1 = relationship is symmetrical (e.g. Related term), 0 = relation is bidirectional (e.g. Is a type of/Has a type)
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL

	Name	Data type	Description / Attributes
grid icon	source_id	int	For internal use only <b>Nullable</b>

Linked from

	Table	Join	Title / Name / Description
grid icon	dbo.glossary_term_relationships (Term Relationships)	dbo.glossary_term_relationship_types.type_id = dbo.glossary_term_relationships.type_id	FK_glossary_term_relationships_glossary_term_relationship_types

Unique keys

	Columns	Name / Description
key icon	type_id	PK_glossary_term_relationship_types

Triggers

	Name	When	Description
lightning icon	trg_glossary_term_relationship_type_s_Modify	After Insert, Update	

#### 2.4.1.3. Table: dbo.glossary\_term\_relationships (Term Relationships)

Term relationships (apart from hierarchy).

Columns

	Name	Data type	Description / Attributes
grid icon	relationship_id	int	Row id <b>Identity / Auto increment</b>
grid icon	source_term_id	int	Relationship source term <b>References:</b> dbo.glossary_terms
grid icon	destination_term_id	int	Relationship destination term <b>References:</b> dbo.glossary_terms
grid icon	type_id	int	Relationship type from the dictionary <b>References:</b> dbo.glossary_term_relationship_types
grid icon	description	nvarchar(MAX)	User relationship comment <b>Nullable</b>
grid icon	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
grid icon	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
grid icon	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
grid icon	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
grid icon	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.glossary_term_relationship_types (Term Relationship Types)	<code>dbo.glossary_term_relationships.type_id = dbo.glossary_term_relationship_types.type_id</code>	FK_glossary_term_relationships_glossary_term_relationship_types
→	dbo.glossary_terms (Business Glossary Terms)	<code>dbo.glossary_term_relationships.destination_term_id = dbo.glossary_terms.term_id</code>	FK_glossary_term_relationships_glossary_terms_destination
→	dbo.glossary_terms (Business Glossary Terms)	<code>dbo.glossary_term_relationships.source_term_id = dbo.glossary_terms.term_id</code>	FK_glossary_term_relationships_glossary_terms_source

## Unique keys

	Columns	Name / Description
🔑	relationship_id	PK_glossary_term_relationships
🔑	source_term_id, destination_term_id, type_id	UK_glossary_term_relationships

## Triggers

	Name	When	Description
⚡	trg_glossary_term_relationships_Modify	After Insert, Update	

## 2.4.1.4. Table: dbo.glossary\_term\_types (Term Types)

Dictionary of term types. Built in are: Term, Category, Rule, Policy.

### Columns

	Name	Data type	Description / Attributes
🔑	term_type_id	int	Row id <b>Identity / Auto increment</b>
☰	title	nvarchar(250)	Term type title/name (Term, Category, Policy, ...)
☰	icon_id	int	Id of built in term icon. Default icon if null. <b>Nullable</b>
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
☰	source_id	int	For internal use only <b>Nullable</b>

## Linked from

	Table	Join	Title / Name / Description
→	dbo.glossary_terms (Business Glossary Terms)	dbo.glossary_term_types.term_type_id = dbo.glossary_terms.type_id	FK_glossary_terms_glossary_term_types

## Unique keys

	Columns	Name / Description
🔑	term_type_id	PK_glossary_term_types

## Triggers

	Name	When	Description
⚡	trg_glossary_term_types_Modify	After Insert, Update	

## 2.4.1.5. Table: dbo.glossary\_terms (Business Glossary Terms)

Holds all terms, categories, policies and other types created in Business Glossary.

### Columns

	Name	Data type	Description / Attributes
🔑	term_id	int	Row id <b>Identity / Auto increment</b>
☰	database_id	int	Foreign key to business glossary term belongs to. <b>References:</b> dbo.databases
☰	type_id	int	Type of term: term, policy, category, etc. <b>References:</b> dbo.glossary_term_types
☰	parent_id	int	Parent term <b>Nullable</b> <b>References:</b> dbo.glossary_terms
☰	title	nvarchar(250)	Title/name of the term
☰	description	nvarchar(MAX)	Term description (body) <b>Nullable</b>
☰	description_search	nvarchar(MAX)	Holds description in plain text and lower case. Column used for searching. <b>Nullable</b>
☰	description_plain	nvarchar(MAX)	Holds description in plain text only, without HTML tags <b>Nullable</b>
☰	field1	nvarchar(MAX)	Custom field 1 <b>Nullable</b>
☰	field2	nvarchar(MAX)	Custom field 2 <b>Nullable</b>
☰	field3	nvarchar(MAX)	Custom field 3 <b>Nullable</b>
☰	field4	nvarchar(MAX)	Custom field 4 <b>Nullable</b>
☰	field5	nvarchar(MAX)	Custom field 5 <b>Nullable</b>
☰	field6	nvarchar(MAX)	Custom field 6 <b>Nullable</b>
☰	field7	nvarchar(MAX)	Custom field 7 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field8		nvarchar(MAX)	Custom field 8 <b>Nullable</b>
■	field9		nvarchar(MAX)	Custom field 9 <b>Nullable</b>
■	field10		nvarchar(MAX)	Custom field 10 <b>Nullable</b>
■	field11		nvarchar(MAX)	Custom field 11 <b>Nullable</b>
■	field12		nvarchar(MAX)	Custom field 12 <b>Nullable</b>
■	field13		nvarchar(MAX)	Custom field 13 <b>Nullable</b>
■	field14		nvarchar(MAX)	Custom field 14 <b>Nullable</b>
■	field15		nvarchar(MAX)	Custom field 15 <b>Nullable</b>
■	field16		nvarchar(MAX)	Custom field 16 <b>Nullable</b>
■	field17		nvarchar(MAX)	Custom field 17 <b>Nullable</b>
■	field18		nvarchar(MAX)	Custom field 18 <b>Nullable</b>
■	field19		nvarchar(MAX)	Custom field 19 <b>Nullable</b>
■	field20		nvarchar(MAX)	Custom field 20 <b>Nullable</b>
■	field21		nvarchar(MAX)	Custom field 21 <b>Nullable</b>
■	field22		nvarchar(MAX)	Custom field 22 <b>Nullable</b>
■	field23		nvarchar(MAX)	Custom field 23 <b>Nullable</b>
■	field24		nvarchar(MAX)	Custom field 24 <b>Nullable</b>
■	field25		nvarchar(MAX)	Custom field 25 <b>Nullable</b>
■	field26		nvarchar(MAX)	Custom field 26 <b>Nullable</b>
■	field27		nvarchar(MAX)	Custom field 27 <b>Nullable</b>
■	field28		nvarchar(MAX)	Custom field 28 <b>Nullable</b>
■	field29		nvarchar(MAX)	Custom field 29 <b>Nullable</b>
■	field30		nvarchar(MAX)	Custom field 30 <b>Nullable</b>
■	field31		nvarchar(MAX)	Custom field 31 <b>Nullable</b>

	Name	Data type	Description / Attributes
■	field32	nvarchar(MAX)	Custom field 32 <b>Nullable</b>
■	field33	nvarchar(MAX)	Custom field 33 <b>Nullable</b>
■	field34	nvarchar(MAX)	Custom field 34 <b>Nullable</b>
■	field35	nvarchar(MAX)	Custom field 35 <b>Nullable</b>
■	field36	nvarchar(MAX)	Custom field 36 <b>Nullable</b>
■	field37	nvarchar(MAX)	Custom field 37 <b>Nullable</b>
■	field38	nvarchar(MAX)	Custom field 38 <b>Nullable</b>
■	field39	nvarchar(MAX)	Custom field 39 <b>Nullable</b>
■	field40	nvarchar(MAX)	Custom field 40 <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	field41	nvarchar(MAX)	Custom field 41 <b>Nullable</b>
■	field42	nvarchar(MAX)	Custom field 42 <b>Nullable</b>
■	field43	nvarchar(MAX)	Custom field 43 <b>Nullable</b>
■	field44	nvarchar(MAX)	Custom field 44 <b>Nullable</b>
■	field45	nvarchar(MAX)	Custom field 45 <b>Nullable</b>
■	field46	nvarchar(MAX)	Custom field 46 <b>Nullable</b>
■	field47	nvarchar(MAX)	Custom field 47 <b>Nullable</b>
■	field48	nvarchar(MAX)	Custom field 48 <b>Nullable</b>
■	field49	nvarchar(MAX)	Custom field 49 <b>Nullable</b>
■	field50	nvarchar(MAX)	Custom field 50 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field51		nvarchar(MAX)	Custom field 51 <b>Nullable</b>
■	field52		nvarchar(MAX)	Custom field 52 <b>Nullable</b>
■	field53		nvarchar(MAX)	Custom field 53 <b>Nullable</b>
■	field54		nvarchar(MAX)	Custom field 54 <b>Nullable</b>
■	field55		nvarchar(MAX)	Custom field 55 <b>Nullable</b>
■	field56		nvarchar(MAX)	Custom field 56 <b>Nullable</b>
■	field57		nvarchar(MAX)	Custom field 57 <b>Nullable</b>
■	field58		nvarchar(MAX)	Custom field 58 <b>Nullable</b>
■	field59		nvarchar(MAX)	Custom field 59 <b>Nullable</b>
■	field60		nvarchar(MAX)	Custom field 60 <b>Nullable</b>
■	field61		nvarchar(MAX)	Custom field 61 <b>Nullable</b>
■	field62		nvarchar(MAX)	Custom field 62 <b>Nullable</b>
■	field63		nvarchar(MAX)	Custom field 63 <b>Nullable</b>
■	field64		nvarchar(MAX)	Custom field 64 <b>Nullable</b>
■	field65		nvarchar(MAX)	Custom field 65 <b>Nullable</b>
■	field66		nvarchar(MAX)	Custom field 66 <b>Nullable</b>
■	field67		nvarchar(MAX)	Custom field 67 <b>Nullable</b>
■	field68		nvarchar(MAX)	Custom field 68 <b>Nullable</b>
■	field69		nvarchar(MAX)	Custom field 69 <b>Nullable</b>
■	field70		nvarchar(MAX)	Custom field 70 <b>Nullable</b>
■	field71		nvarchar(MAX)	Custom field 71 <b>Nullable</b>
■	field72		nvarchar(MAX)	Custom field 72 <b>Nullable</b>
■	field73		nvarchar(MAX)	Custom field 73 <b>Nullable</b>
■	field74		nvarchar(MAX)	Custom field 74 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field75		nvarchar(MAX)	Custom field 75 <b>Nullable</b>
■	field76		nvarchar(MAX)	Custom field 76 <b>Nullable</b>
■	field77		nvarchar(MAX)	Custom field 77 <b>Nullable</b>
■	field78		nvarchar(MAX)	Custom field 78 <b>Nullable</b>
■	field79		nvarchar(MAX)	Custom field 79 <b>Nullable</b>
■	field80		nvarchar(MAX)	Custom field 80 <b>Nullable</b>
■	field81		nvarchar(MAX)	Custom field 81 <b>Nullable</b>
■	field82		nvarchar(MAX)	Custom field 82 <b>Nullable</b>
■	field83		nvarchar(MAX)	Custom field 83 <b>Nullable</b>
■	field84		nvarchar(MAX)	Custom field 84 <b>Nullable</b>
■	field85		nvarchar(MAX)	Custom field 85 <b>Nullable</b>
■	field86		nvarchar(MAX)	Custom field 86 <b>Nullable</b>
■	field87		nvarchar(MAX)	Custom field 87 <b>Nullable</b>
■	field88		nvarchar(MAX)	Custom field 88 <b>Nullable</b>
■	field89		nvarchar(MAX)	Custom field 89 <b>Nullable</b>
■	field90		nvarchar(MAX)	Custom field 90 <b>Nullable</b>
■	field91		nvarchar(MAX)	Custom field 91 <b>Nullable</b>
■	field92		nvarchar(MAX)	Custom field 92 <b>Nullable</b>
■	field93		nvarchar(MAX)	Custom field 93 <b>Nullable</b>
■	field94		nvarchar(MAX)	Custom field 94 <b>Nullable</b>
■	field95		nvarchar(MAX)	Custom field 95 <b>Nullable</b>
■	field96		nvarchar(MAX)	Custom field 96 <b>Nullable</b>
■	field97		nvarchar(MAX)	Custom field 97 <b>Nullable</b>
■	field98		nvarchar(MAX)	Custom field 98 <b>Nullable</b>

		Name	Data type	Description / Attributes
■	field99		nvarchar(MAX)	Custom field 99 <b>Nullable</b>
■	field100		nvarchar(MAX)	Custom field 100 <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	<b>dbo.glossary_terms.database_id = dbo.databases.database_id</b>	FK_glossary_terms_database
→	dbo.glossary_term_types (Term Types)	<b>dbo.glossary_terms.type_id = dbo.glossary_term_types.term_type_id</b>	FK_glossary_terms_glossary_term_types
→	dbo.glossary_terms (Business Glossary Terms)	<b>dbo.glossary_terms.parent_id = dbo.glossary_terms.term_id</b>	FK_glossary_terms_glossary_terms

## Linked from

	Table	Join	Title / Name / Description
←	dbo.glossary_terms (Business Glossary Terms)	<b>dbo.glossary_terms.term_id = dbo.glossary_terms.parent_id</b>	FK_glossary_terms_glossary_terms
←	dbo.glossary_mappings (Term-Data links)	<b>dbo.glossary_terms.term_id = dbo.glossary_mappings.term_id</b>	FK_glossary_mappings_glossary_terms
←	dbo.glossary_term_relationships (Term Relationships)	<b>dbo.glossary_terms.term_id = dbo.glossary_term_relationships.destination_term_id</b>	FK_glossary_term_relationships_glossary_terms_destination
←	dbo.glossary_term_relationships (Term Relationships)	<b>dbo.glossary_terms.term_id = dbo.glossary_term_relationships.source_term_id</b>	FK_glossary_term_relationships_glossary_terms_source

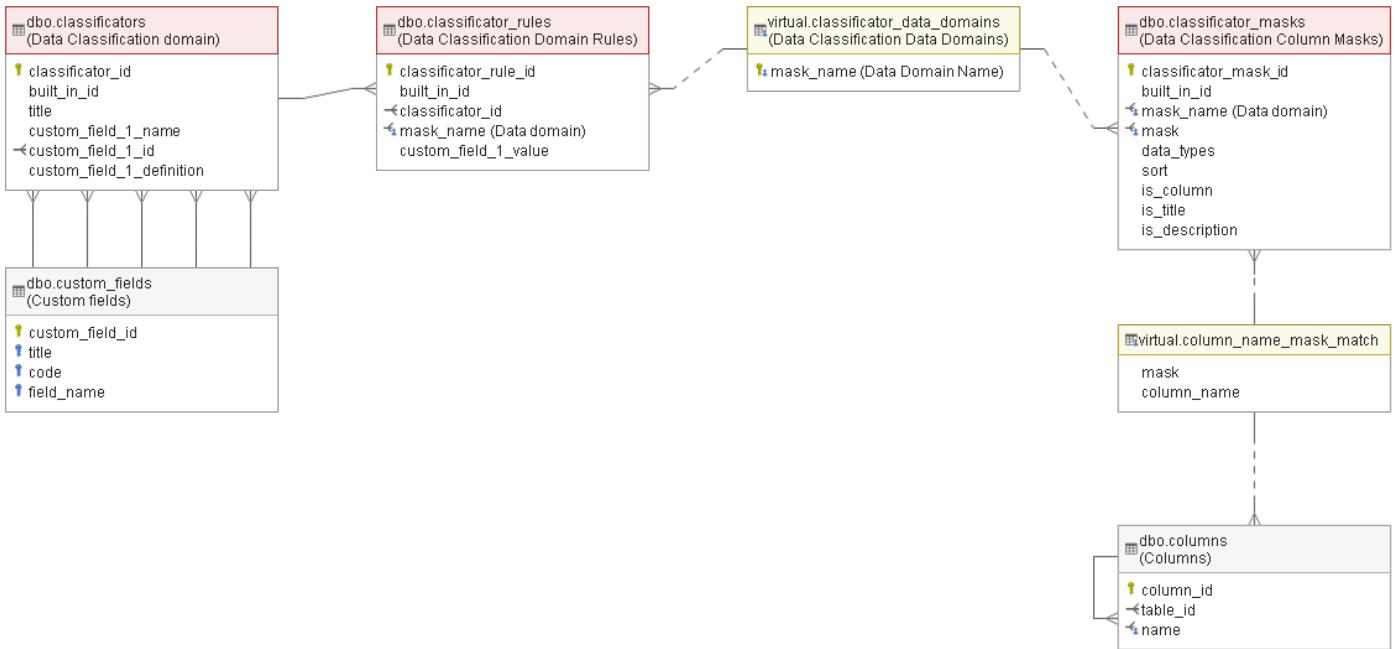
## Unique keys

	Columns	Name / Description
🔑	term_id	PK_glossary_terms
🔑	title	UK_glossary_terms_title

## Triggers

	Name	When	Description
⚡	trg_glossary_terms_Modify	After Insert, Update	

## 2.5. Data Classification



Tables used by Data Classification function that searches for table columns based on their names, titles, description, linked glossary terms.

Data classification in Dataedo is performed independently for "classification domains" (**classifiers** table). Examples of domains are GDPR, PII.

Each domain has a list of rules (**classifier\_rules** table), i.e. what "data domains" (e.g. email, bank account) should be included and what classification labels (e.g. Non-Sensitive PII) should be suggested.

Column search is performed based on predefined masks (**classifier\_masks** table) for each data domain. For instance *Date of birth* has following masks: '%date\_of\_birth%', '%dateofbirth%', '%birth%date%'.

### 2.5.1. Tables

#### 2.5.1.1. Table: dbo.classifier\_masks (Data Classification Column Masks)

List of SQL like masks for column names, titles (and optionally description) used in search of columns potentially holding sensitive data. For instance '%date%birth%'. Masks are grouped into "data domains" (with **mask\_name** field) representing one type of data (e.g. email, bank account, etc.).

##### Columns

	Name	Data type	Description / Attributes
KEY	classifier_mask_id	int	Row id <b>Identity / Auto increment</b>
	built_in_id	int	Global static ID of the mask delivered by Dataedo. Should be null for user entries. Doesn't change from installation to installation, regardless of the order of inserting rows and into database. <b>Nullable</b>
	mask_name (Data domain)	nvarchar(250)	Name of data domain, e.g. bank account or email <b>References:</b> virtual.classifier_data_domains
	mask	nvarchar(250)	SQL like mask used to find column names. For instance '%date%birth%'. <b>References:</b> virtual.column_name_mask_match

	Name	Data type	Description / Attributes
■	data_types	nvarchar(250)	Columns with what data types (grouped) should be considered, i.e. text, number, date. Types can be listed separated with space and coma. By default null and means that all data types should be considered. Available data types: text, number, date <b>Nullable</b>
■	sort	int	Order in which masks should be considered to establish data domain. The one with the least number is selected. <b>Default:</b> 99
■	is_column	bit	Should mask be used for column names. 1 - include, 0 - ignore <b>Default:</b> 1
■	is_title	bit	Should mask be used for column titles. 1 - include, 0 - ignore <b>Default:</b> 0
■	is_description	bit	Should mask be used for column description. 1 - include, 0 - ignore <b>Default:</b> 0
■	comments	nvarchar(MAX)	Comments for person defining masks <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→ i	virtual.classifier_data_domains (Data Classification Data Domains)	<b>dbo.classifier_masks.mask_name = virtual.classifier_data_domains.mask_name</b>	fk_classifier_mask_name_classifier_masks
→ i	virtual.column_name_mask_match	<b>dbo.classifier_masks.mask = virtual.column_name_mask_match.mask</b>	fk_column_name_mask_match_classifier_masks

## Unique keys

	Columns	Name / Description
🔑	classifier_mask_id	PK_classifier_masks

## Triggers

	Name	When	Description
⚡	trg_classifier_masks_Modify	After Insert, Update	

## 2.5.1.2. Table: dbo.classifier\_rules (Data Classification Domain Rules)

Table defines holds a list of rules for each classification domain. i.e. what "data domains" (e.g. email, bank account) should be included and what classification labels (e.g. Non-Sensitive PII) should be suggested.

## Columns

	Name	Data type	Description / Attributes
PK	classifier_rule_id	int	Row id <b>Identity / Auto increment</b>
PK	built_in_id	int	Global static ID of the domain rule delivered by Dataedo. Should be null for user entries. Doesn't change from installation to installation, regardless of the order of inserting rows and into database. <b>Nullable</b>
PK	classifier_id	int	Rule classification domain <b>References:</b> dbo.classifiers
PK	mask_name (Data domain)	nvarchar(250)	Name of the data domain that should be searched. <b>References:</b> virtual.classifier_data_domains
	custom_field_1_value	nvarchar(250)	Predefined classification/domain label suggestion for classification domain field #1 <b>Nullable</b>
	custom_field_2_value	nvarchar(250)	Predefined classification/domain label suggestion for classification domain field #2 <b>Nullable</b>
	custom_field_3_value	nvarchar(250)	Predefined classification/domain label suggestion for classification domain field #3 <b>Nullable</b>
	custom_field_4_value	nvarchar(250)	Predefined classification/domain label suggestion for classification domain field #4 <b>Nullable</b>
	custom_field_5_value	nvarchar(250)	Predefined classification/domain label suggestion for classification domain field #5 <b>Nullable</b>
	comments	nvarchar(MAX)	Comments for person defining rules <b>Nullable</b>
	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	virtual.classifier_data_domains (Data Classification Data Domains)	dbo.classifier_rules.mask_name = virtual.classifier_data_domains.mask_name	fk_classifier_mask_name_classifier_rules
→	dbo.classifiers (Data Classification domain)	dbo.classifier_rules.classifier_id = dbo.classifiers.classifier_id	FK_classifier_rules_classifiers

## Unique keys

	Columns	Name / Description
🔑	classifier_rule_id	PK_classifier_rules

## Triggers

	Name	When	Description
⚡	trg_classifier_rules_Modify	After Insert, Update	

### 2.5.1.3. Table: dbo.classifiers (Data Classification domain)

Table holds Data Classification domains. Domain is a distinct approach to classification with its separate classification fields and classification labels. Example domains: *GDPR*, *PII*.

Each domain has 1-3 custom fields that are used for classification labels. For instance in **PII** domain:

- **PII** - Non-Sensitive PII, Sensitive PII, Non-PII
- **PII Data Domain** - Address, Name, etc.

## Columns

	Name	Data type	Description / Attributes
🔑	classifier_id	int	Row id <b>Identity / Auto increment</b>
☰	built_in_id	int	Global static ID of the classification domain delivered by Dataedo. Should be null for user entries. Doesn't change from installation to installation, regardless of the order of inserting rows and into database. <b>Nullable</b>
☰	title	nvarchar(250)	Data Classification domain name, e.g. GDPR
☰	custom_field_1_name	nvarchar(250)	Name of the custom field that will be created for classification field #1 <b>Nullable</b>
☰	custom_field_2_name	nvarchar(250)	Name of the custom field that will be created for classification field #2 <b>Nullable</b>
☰	custom_field_3_name	nvarchar(250)	Name of the custom field that will be created for classification field #3 <b>Nullable</b>
☰	custom_field_4_name	nvarchar(250)	Name of the custom field that will be created for classification field #4 <b>Nullable</b>
☰	custom_field_5_name	nvarchar(250)	Name of the custom field that will be created for classification field #5 <b>Nullable</b>
☰	custom_field_1_id	int	Foreign key to custom field of classification field #1 <b>Nullable</b> References: dbo.custom_fields
☰	custom_field_2_id	int	Foreign key to custom field of classification field #2 <b>Nullable</b> References: dbo.custom_fields
☰	custom_field_3_id	int	Foreign key to custom field of classification field #3 <b>Nullable</b> References: dbo.custom_fields
☰	custom_field_4_id	int	Foreign key to custom field of classification field #4 <b>Nullable</b> References: dbo.custom_fields
☰	custom_field_5_id	int	Foreign key to custom field of classification field #5 <b>Nullable</b> References: dbo.custom_fields

	Name	Data type	Description / Attributes
■	custom_field_1_definition	nvarchar(MAX)	List of values for classification field #1 that will be written into custom_fields.definition field <b>Nullable</b>
■	custom_field_2_definition	nvarchar(MAX)	List of values for classification field #2 that will be written into custom_fields.definition field <b>Nullable</b>
■	custom_field_3_definition	nvarchar(MAX)	List of values for classification field #3 that will be written into custom_fields.definition field <b>Nullable</b>
■	custom_field_4_definition	nvarchar(MAX)	List of values for classification field #4 that will be written into custom_fields.definition field <b>Nullable</b>
■	custom_field_5_definition	nvarchar(MAX)	List of values for classification field #5 that will be written into custom_fields.definition field <b>Nullable</b>
■	custom_field_1_class_id	int	Default class of the custom field that will be created for classification field #1 <b>Nullable</b> <b>References:</b> dbo.custom_field_classes
■	custom_field_2_class_id	int	Default class of the custom field that will be created for classification field #2 <b>Nullable</b> <b>References:</b> dbo.custom_field_classes
■	custom_field_3_class_id	int	Default class of the custom field that will be created for classification field #3 <b>Nullable</b> <b>References:</b> dbo.custom_field_classes
■	custom_field_4_class_id	int	Default class of the custom field that will be created for classification field #4 <b>Nullable</b> <b>References:</b> dbo.custom_field_classes
■	custom_field_5_class_id	int	Default class of the custom field that will be created for classification field #5 <b>Nullable</b> <b>References:</b> dbo.custom_field_classes
■	description	nvarchar(MAX)	Description of classification domain <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.custom_field_classes (Field class)	<b>dbo.classifiers.custom_field_1_class_id</b> = dbo.custom_field_classes.custom_field_class_id	FK_classifiers_custom_fields_classes_1
→	dbo.custom_field_classes (Field class)	<b>dbo.classifiers.custom_field_2_class_id</b> = dbo.custom_field_classes.custom_field_class_id	FK_classifiers_custom_fields_classes_2
→	dbo.custom_field_classes (Field class)	<b>dbo.classifiers.custom_field_3_class_id</b> = dbo.custom_field_classes.custom_field_class_id	FK_classifiers_custom_fields_classes_3
→	dbo.custom_field_classes (Field class)	<b>dbo.classifiers.custom_field_4_class_id</b> = dbo.custom_field_classes.custom_field_class_id	FK_classifiers_custom_fields_classes_4
→	dbo.custom_field_classes (Field class)	<b>dbo.classifiers.custom_field_5_class_id</b> = dbo.custom_field_classes.custom_field_class_id	FK_classifiers_custom_fields_classes_5
→	dbo.custom_fields (Custom fields)	<b>dbo.classifiers.custom_field_1_id</b> = dbo.custom_fields.custom_field_id	FK_classifiers_custom_fields_1
→	dbo.custom_fields (Custom fields)	<b>dbo.classifiers.custom_field_2_id</b> = dbo.custom_fields.custom_field_id	FK_classifiers_custom_fields_2
→	dbo.custom_fields (Custom fields)	<b>dbo.classifiers.custom_field_3_id</b> = dbo.custom_fields.custom_field_id	FK_classifiers_custom_fields_3
→	dbo.custom_fields (Custom fields)	<b>dbo.classifiers.custom_field_4_id</b> = dbo.custom_fields.custom_field_id	FK_classifiers_custom_fields_4
→	dbo.custom_fields (Custom fields)	<b>dbo.classifiers.custom_field_5_id</b> = dbo.custom_fields.custom_field_id	FK_classifiers_custom_fields_5

## Linked from

	Table	Join	Title / Name / Description
←	dbo.classifier_rules (Data Classification Domain Rules)	<b>dbo.classifiers.classifier_id</b> = dbo.classifier_rules.classifier_id	FK_classifier_rules_classifiers

## Unique keys

	Columns	Name / Description
🔑	classifier_id	PK_classifiers

## Triggers

	Name	When	Description
⚡	trg_classifiers_Modify	After Insert, Update	

### 2.5.1.4. Table: virtual.classifier\_data\_domains (Data Classification Data Domains)

This is a virtual table representing "data domain", i.e. one type of data such as email, address, bank account, etc.. In Dataedo repository it has no lookup table and is identified by the name kept int **mask\_name** field.

## Columns

		Name	Data type	Description / Attributes
		mask_name (Data Domain Name)	nvarchar	Data domain name

## Linked from

	Table	Join	Title / Name / Description
	dbo.classifier_masks (Data Classification Column Masks)	<code>virtual.classifier_data_domains.mask_name = dbo.classifier_masks.mask_name</code>	fk_classifier_mask_name_classifier_masks
	dbo.classifier_rules (Data Classification Domain Rules)	<code>virtual.classifier_data_domains.mask_name = dbo.classifier_rules.mask_name</code>	fk_classifier_mask_name_classifier_rules

## Unique keys

	Columns	Name / Description
	mask_name	pk_classificatior_mask_name_mask_name

## 2.5.1.5. Table: virtual.column\_name\_mask\_match

This is a virtual table that represents many-to-many join between **dbo.classifier\_masks** and **dbo.columns**. Columns are matched in the following way:

### **column.name like mask**

For instance:

'Date of birth' like '%date%birth%'

## Columns

		Name	Data type	Description / Attributes
		mask	nvarchar	Column match mask, e.g. '%date%birth%'
		column_name	nvarchar	Name of the column

## Linked from

	Table	Join	Title / Name / Description
	dbo.classifier_masks (Data Classification Column Masks)	<code>virtual.column_name_mask_match.mask = dbo.classifier_masks.mask</code>	fk_column_name_mask_match_classifier_masks
	dbo.columns (Columns)	<code>virtual.column_name_mask_match.column_name = dbo.columns.name</code>	fk_column_name_mask_match_columns

## 2.6. Dependencies

Structure for storing object dependencies.

### 2.6.1. Tables

#### 2.6.1.1. Table: dbo.dependencies (Dependencies)

Object dependencies imported from source database.

It links two objects of various types - tables, views, procedures, functions and triggers. There is *referencing* object and *referenced* object. *Referencing* object depends on *referenced* object.

Because those objects may not be imported to repository or exist in other databases foreign and primary keys are not used. A combination of fields is used to identify object: server, database, schema, name and type.

Please note that unlike most of other tables, rows in this one are not related to any specific documentation/database.

Columns

	Name	Data type	Description / Attributes
💡	dependency_id	int	Row id <b>Identity / Auto increment</b>
💡	referencing_server	nvarchar(250)	Referencing object server name <b>Nullable</b>
💡	referencing_database	nvarchar(250)	Referencing object database name <b>Nullable</b>
💡	referencing_schema	nvarchar(250)	Referencing object schema name <b>Nullable</b>
💡	referencing_name	nvarchar(250)	Referencing object name
💡	referencing_type	nvarchar(50)	Referencing object type: TABLE, VIEW, PROCEDURE, FUNCTION, TRIGGER
💡	referenced_server	nvarchar(250)	Referenced object server name <b>Nullable</b>
💡	referenced_database	nvarchar(250)	Referenced object database name <b>Nullable</b>
💡	referenced_schema	nvarchar(250)	Referenced object schema name <b>Nullable</b>
💡	referenced_name	nvarchar(250)	Referenced object name
💡	referenced_type	nvarchar(50)	Referenced object type: TABLE, VIEW, PROCEDURE, FUNCTION, TRIGGER <b>Nullable</b>
💡	source	nvarchar(50)	Source of the row: 'DBMS' for imported from database, 'USER' for manual defined by user in Dataedo repository <b>Default:</b> 'DBMS'
💡	status	char(1)	Dependency status: A - active, D - deleted. Currently not used. <b>Default:</b> 'A'
💡	is_caller_dependent	char(1)	Field imported from DBMS <b>Nullable</b>
💡	is_ambiguous	char(1)	Field imported from DBMS <b>Nullable</b>
💡	dependency_type	nvarchar(10)	Dependency type imported from DBMS <b>Nullable</b>

	Name	Data type	Description / Attributes
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>
■	temp_sync_status	bit	For internal use only <b>Default:</b> 0

## Unique keys

	Columns	Name / Description
🔑	dependency_id	PK_dependencies Table primary key

## Triggers

	Name	When	Description
⚡	trg_dependencies_Modify	After Insert, Update	

### 2.6.1.2. Table: dbo.dependencies\_descriptions (Dependencies descriptions)

Descriptions of dependencies elements provided in Dataedo repository.

In contrast with dependencies, dependency descriptions are related to a specific documentation/database. Same dependency may have different descriptions if dependency is between objects in two different databases.

## Columns

	Name	Data type	Description / Attributes
■	dependency_descriptions_id	int	Row id <b>Identity / Auto increment</b>
■	database_id	int	Documentation which description is part of <b>References:</b> dbo.databases
■	referencing_server	nvarchar(250)	Referencing object server name <b>Nullable</b>
■	referencing_database	nvarchar(250)	Referencing object database name <b>Nullable</b>
■	referencing_schema	nvarchar(250)	Referencing object schema name <b>Nullable</b>
■	referencing_name	nvarchar(250)	Referencing object name
■	referencing_type	nvarchar(50)	Referencing object type
■	referenced_server	nvarchar(250)	Referenced object server name <b>Nullable</b>
■	referenced_database	nvarchar(250)	Referenced object database name <b>Nullable</b>

	Name	Data type	Description / Attributes
■	referenced_schema	nvarchar(250)	Referenced object schema name <b>Nullable</b>
■	referenced_name	nvarchar(250)	Referenced object name
■	referenced_type	nvarchar(50)	Referenced object type <b>Nullable</b>
■	description	nvarchar(MAX)	Description provided by user in repository <b>Nullable</b>
■	ordinal_position_uses	int	Position used in sorting elements on dependencies list in a tree in uses tab <b>Nullable</b>
■	ordinal_position_used_by	int	Position used in sorting elements on dependencies list in a tree in used by tab <b>Nullable</b>
■	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
■	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
■	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
■	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
■	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.databases (Documentations)	dbo.dependencies_descriptions.database_id = dbo.databases.database_id	FK_dependencies_descriptions_databases

## Unique keys

	Columns	Name / Description
🔑	dependency_descriptions_id	PK_dependencies_descriptions Table primary key

## Triggers

	Name	When	Description
⚡	trg_dependencies_descriptions_Mo dify	After Insert, Update	

## 2.7. Custom fields



Custom fields are additional descriptive fields that you can add to:

- Tables, Views
- Columns
- Procedures, Functions
- Parameters
- Triggers
- Relationship (foreign key)
- Primary/unique key
- Modules

You can define up to 40 fields in repository.

Fields are defined in **custom\_fields** table and for each you can define for which objects should tyeh be available for (e.g. tables, columns). Their definition is repository-wide.

 Custom field: Type

Title  
Type

Type  
Drop-down list (closed) - accepts only one value from a defined list

Definition (type in elements separated with ',')  
Schema, Metadata, Internal, Deprecated, Connection, Other

Visibility  
 Documentation    Module    Table/View    Column    Relation  
 Key    Trigger    Proc/Function    Parameter

Description  
- Schema - schema objects  
- Metadata - additional metadata stored by Dataedo  
- Internal - for internal use only  
- Deprecated - not used anymore

Field: field1

## 2.7.1. Tables

### 2.7.1.1. Table: dbo.custom\_fields (Custom fields)

Table holds definition of custom fields - metadata fields defined by Dataedo admin.

**Custom field: Type**

Title  
Type Field name

Type Field type  
Drop-down list (closed) - accepts only one value from a defined list

Definition (type in elements separated with ',')  
Schema, Metadata, Internal, Deprecated, Connection, Other

List of predefined/available values

Visibility Availability definition  
 Documentation    Module    Table/View    Column    Relation  
 Key    Trigger    Proc/Function    Parameter

Description Admin description  
 - Schema - schema objects  
 - Metadata - additional metadata stored by Dataedo  
 - Internal - for internal use only  
 - Deprecated - not used anymore

Field: field1

OK Cancel

## Availability

Availability for field is stored in following bit columns:

- table\_visibility
- procedure\_visibility
- column\_visibility
- relation\_visibility
- key\_visibility
- trigger\_visibility
- parameter\_visibility
- module\_visibility
- documentation\_visibility

## Field table column

Custom fields values are saved in the same tables as objects they are assigned to, e.g. fields in for **Parameters** are saved in **dbo.parameters** table in columns **field1** to **field40**.

Below example shows how to find physical values in Dataedo repository tables.

Imagine **Sensitive Data** custom field:

**Table: Person.Person**

Table	Columns	Relations	Unique keys	Triggers	Depende
	Key	#	Name	Data type	Sensitive Data
		1	BusinessEntityID	int	No
		2	PersonType	nchar(2)	No
		3	NameStyle	NameStyle: bit	No
		4	Title	nvarchar(8)	No
		5	FirstName	Name: nvarchar(50)	Yes
		6	MiddleName	Name: nvarchar(50)	Yes
		7	LastName	Name: nvarchar(50)	Yes
		8	Suffix	nvarchar(10)	No
		9	EmailPromotion	int	No
		10	AdditionalContactInfo	xml	Yes
		11	Demographics	xml	No
		12	rowguid	uniqueidentifier	No
		13	ModifiedDate	datetime	No

You can find where this value is stored in column **field\_name** in field form or in **custom\_fields** table. In this case it is **field6**.

 New custom field

— □ ×

**Title**  
Sensitive Data

**Type**  
Drop-down list (closed) - accepts only one value from a defined list

**Definition (type in elements separated with ',')**  
Yes, No

**Visibility**

<input type="checkbox"/> Documentation	<input type="checkbox"/> Module	<input checked="" type="checkbox"/> Table/View	<input checked="" type="checkbox"/> Column	<input type="checkbox"/> Relation
<input type="checkbox"/> Key	<input type="checkbox"/> Trigger	<input type="checkbox"/> Proc/Function	<input type="checkbox"/> Parameter	

**Description**

Field: field6

OK Cancel

```

SELECT [custom_field_id]
      ,[ordinal_position]
      ,[title]
      ,[field_name]
  FROM [dbo].[custom_fields]

```

100 %

Results Messages

	custom_field_id	ordinal_position	title	field_name
1	16	1	Data Owner	field3
2	17	2	Data Steward	field4
3	19	3	Sensitive Data	field6
4	20	4	Source System	field7
5	21	5	Source Table	field8
6	22	6	Source Column	field9
7	23	7	Transformation	field10
8	24	8	Format	field11
9	25	9	Application Form	field12
10	26	10	Application Form Field	field13
11	29	11	Status	field16

So to find Sensitive Data values in columns you need to look into **field6** column in **columns** table.

```

SELECT [column_id]
      ,[table_id]
      ,[ordinal_position]
      ,[name]
      ,[datatype]
      ,[field6]
  FROM [dbo].[columns]
 where table_id = 144

```

100 %

Results Messages

	column_id	table_id	ordinal_position	name	datatype	field6
1	40713	144	1	BusinessEntityID	int	No
2	40714	144	2	PersonType	nchar(2)	No
3	40715	144	3	NameStyle	NameStyle: bit	No
4	40716	144	4	Title	nvarchar(8)	No
5	40717	144	5	FirstName	Name: nvarchar(50)	Yes
6	40718	144	6	MiddleName	Name: nvarchar(50)	Yes
7	40719	144	7	LastName	Name: nvarchar(50)	Yes
8	40720	144	8	Suffix	nvarchar(10)	No
9	40721	144	9	EmailPromotion	int	No
10	40722	144	10	AdditionalContactInfo	xml	Yes
11	40723	144	11	Demographics	xml	No
12	40724	144	12	rowguid	uniqueidentifier	No
13	40725	144	13	ModifiedDate	datetime	No

## Columns

	Name	Data type	Description / Attributes
custom_field_id	custom_field_id	int	Row id <b>Identity / Auto increment</b>
title	title	nvarchar(250)	Field displayed name
type	type	nvarchar(100)	Field type: TEXT - plain text field, LIST_CLOSED - closed drop-down LIST_OPEN - open drop-down MULTI_VALUE_LIST_CLOSED - closed multi-select drop-down INTEGER - number CHECKBOX - checkbox <b>Default:</b> N'TEXT'
code	code	nvarchar(255)	Text code set on creation of the field
field_name	field_name	nvarchar(50)	Name of the column in physical tables where custom field is stored, e.g. field1 (which is in all schema tables, i.e. tables.field1, columns.field1...) <b>Nullable</b>
definition	definition	nvarchar(MAX)	List of values for lists as text separated with commas. E.g. 'Blue, Red, Green' <b>Nullable</b>
ordinal_position	ordinal_position	int	User sorting column
description	description	nvarchar(MAX)	Custom field admin comments <b>Nullable</b>
table_visibility	table_visibility	bit	Is field available for tables and views (columns are defined separately) <b>Default:</b> 1
procedure_visibility	procedure_visibility	bit	Is field available for procedures and functions (columns are defined separately) <b>Default:</b> 1
column_visibility	column_visibility	bit	Is field available for tables' and views' columns <b>Default:</b> 1
relation_visibility	relation_visibility	bit	Is field available for table/view relations <b>Default:</b> 1
key_visibility	key_visibility	bit	Is field available for table/view primary/unique keys <b>Default:</b> 1
trigger_visibility	trigger_visibility	bit	Is field available for triggers <b>Default:</b> 1
parameter_visibility	parameter_visibility	bit	Is table available for parameters <b>Default:</b> 1
module_visibility	module_visibility	bit	Is field available for modules <b>Default:</b> 1
documentation_visibility	documentation_visibility	bit	Is field available for documentation <b>Default:</b> 1
term_visibility	term_visibility	bit	Is field available for business glossary term <b>Default:</b> 1
creation_date	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
created_by	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
last_modification_date	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()

	Name	Data type	Description / Attributes
[[table]]	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
[[table]]	source_id	int	For internal use only <b>Nullable</b>
[[table]]	custom_field_class_id	int	Global class (type) of custom field (e.g. Status, Classification) <b>References:</b> dbo.custom_field_classes

## Links to

	Table	Join	Title / Name / Description
→	dbo.custom_field_classes (Field class)	<b>dbo.custom_fields.custom_field_class_id = dbo.custom_field_classes.custom_field_class_id</b>	FK_custom_fields_custom_field_classes

## Linked from

	Table	Join	Title / Name / Description
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_fields.custom_field_id = dbo.classifiers.custom_field_1_id</b>	FK_classifiers_custom_fields_1
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_fields.custom_field_id = dbo.classifiers.custom_field_2_id</b>	FK_classifiers_custom_fields_2
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_fields.custom_field_id = dbo.classifiers.custom_field_3_id</b>	FK_classifiers_custom_fields_3
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_fields.custom_field_id = dbo.classifiers.custom_field_4_id</b>	FK_classifiers_custom_fields_4
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_fields.custom_field_id = dbo.classifiers.custom_field_5_id</b>	FK_classifiers_custom_fields_5
→	dbo.custom_fields_values (Custom fields index)	<b>dbo.custom_fields.custom_field_id = dbo.custom_fields_values.custom_field_id</b>	FK_custom_fields_values_custom_fields
→	dbo.documentation_custom_fields (Documentation custom fields)	<b>dbo.custom_fields.custom_field_id = dbo.documentation_custom_fields.custom_field_id</b>	FK_documentation_custom_fields_custom_fields

## Unique keys

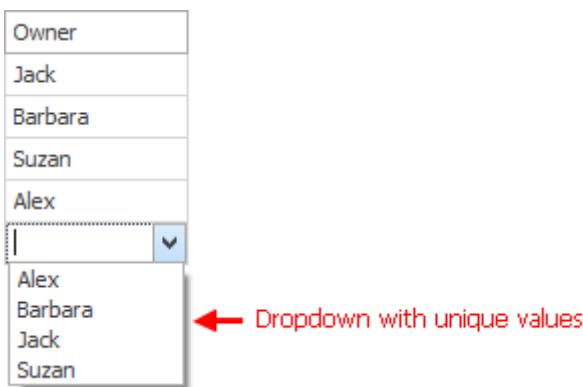
	Columns	Name / Description
🔑	custom_field_id	PK_custom_fields Table primary key
🔑	code	UK_custom_fields_code Code must be unique
🔑	field_name	UK_custom_fields_field_name Any physical column in Dataedo repository that holds custom fields must obviously be used just by one field
🔑	title	UK_custom_fields_title Field name must be unique

## Triggers

	Name	When	Description
⚡	trg_custom_fields_Modify	After Insert, Update	

### 2.7.1.2. Table: dbo.custom\_fields\_values (Custom fields index)

This is an indexing table to efficiently create a dropdown for **open lists** (lists that user can add values to in each cell).



#### Values

This index table holds value in each cell in each row of each open list custom field.

#### Cell identifier

To identify specific cell that this value refers to you need to use following columns:

1. custom\_field\_id
2. object\_type - type of the object that value was entered for e.g. COLUMN
3. object\_id - id of this object

	id	custom_field_id	object_type	object_id	value
1	11	6	COLUMN	32541	Jack
2	12	6	COLUMN	32542	Barbara
3	13	6	COLUMN	32543	Suzan
4	14	6	COLUMN	32544	Alex

#### Populating table

Dataedo writes rows to this table:

1. Each time you edit object
2. When type of custom field is changed to open list

#### Listing distinct values

If you want to list all values used in the repository you can do it with this query:

```
SELECT DISTINCT [value]
  FROM dbo.custom_fields_values
 WHERE custom_field_id = 4
```

#### Columns

		Name	Data type	Description / Attributes
■	🔑	id	int	Row id Identity / Auto increment
■	🔑	custom_field_id	int	Custom field References: dbo.custom_fields
■	🔑	object_type	nvarchar(100)	Code of object type: TABLE, VIEW, PROCEDURE, FUNCTION, COLUMN, PARAMETER, KEY, RELATION, TRIGGER, MODULE, DATABASE
■	🔑	object_id	int	Id of the relevant object type (table, column, etc.)
■		value	nvarchar(100)	Value of the cell

## Links to

	Table	Join	Title / Name / Description
→	dbo.custom_fields (Custom fields)	<b>dbo.custom_fields_values.custom_field</b> _id = dbo.custom_fields.custom_field_id	FK_custom_fields_values_custom_fields

## Unique keys

	Columns	Name / Description
🔑	id	PK_custom_fields_values Table primary key
🔑	custom_field_id, object_type, object_id	Cell identifier Cell identifier

## 2.7.1.3. Table: dbo.documentation\_custom\_fields (Documentation custom fields)

Table holds mapping of **custom fields** to **extended properties** (SQL Server/Azure SQL). Custom field can be mapped to different extended property in each database.

### Columns

	Name	Data type	Description / Attributes
☰	documentation_custom_field_id	int	Row id <b>Identity / Auto increment</b>
☰	custom_field_id	int	Ref: custom_fields <b>References:</b> dbo.custom_fields
☰	database_id	int	Ref: databases <b>References:</b> dbo.databases
☰	extended_property	nvarchar(128)	Name of the extended property
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
☰	source_id	int	For internal use only <b>Nullable</b>

## Links to

	Table	Join	Title / Name / Description
→	dbo.custom_fields (Custom fields)	<b>dbo.documentation_custom_fields.cus</b> tom_field_id = dbo.custom_fields.custom_field_id	FK_documentation_custom_fields_custom_fields
→	dbo.databases (Documentations)	<b>dbo.documentation_custom_fields.dat</b> abase_id = dbo.databases.database_id	FK_documentation_custom_fields_databases

## Unique keys

	Columns	Name / Description
	documentation_custom_field_id	PK_documentation_custom_fields Table primary key
	custom_field_id, database_id	UK_documentation_custom_fields Field can be mapped only once per connection

## Triggers

	Name	When	Description
	trg_documentation_custom_fields_Modify	After Insert, Update	

## 2.8. Utils

Utility tables used by Dataedo.

### 2.8.1. Tables

#### 2.8.1.1. Table: dbo.guid (Repo ID)

Holds globally unique identifier of the repository. Holds one row inserted at a time of the creation of the repository.

Columns

		Name	Data type	Description / Attributes
	guid		uniqueidentifier	Unique identifier of the repository, e.g. 7AE2EF52-B417-4219-B42F-12F63D8755F0 <b>Default:</b> newsequentialid()

Unique keys

	Columns	Name / Description
	guid	PK_guid Table primary key

#### 2.8.1.2. Table: dbo.ignored\_objects (Ignored objects)

Objects ignored from import from DBMS.

Columns

		Name	Data type	Description / Attributes
	ignored_object_id		int	Row id <b>Identity / Auto increment</b>
	database_id		int	Documentation from which object was ignored <b>References:</b> dbo.databases
	schema		nvarchar(250)	Schema name <b>Nullable</b>
	name		nvarchar(250)	Database object name
	object_type		nvarchar(100)	Type of database object: TABLE/VIEW/PROCEDURE/FUNCTION
	creation_date		datetime	Standard row creation timestamp <b>Default:</b> getdate()
	created_by		nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
	last_modification_date		datetime	Standard row last update timestamp <b>Default:</b> getdate()
	modified_by		nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
	source_id		int	For internal use only <b>Nullable</b>

Links to

	Table	Join	Title / Name / Description
	dbo.databases (Documentations)	dbo.ignored_objects.database_id = dbo.databases.database_id	FK_ignored_objects_databases

## Unique keys

	Columns	Name / Description
🔑	ignored_object_id	PK_ignored_objects Table primary key

## Triggers

	Name	When	Description
⚡	trg_ignored_objects_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.8.1.3. Table: dbo.licenses (Users)

User accounts and corresponding license keys.

### Columns

	Name	Data type	Description / Attributes
☰	license_id	int	Row id <b>Identity / Auto increment</b>
☰	login	nvarchar(1024)	Dataedo user login. This matches repository database login.
☰	key	varchar(256)	License key. That includes Free license key. This entry is further encrypted.
☰	host	nvarchar(50)	Name of host that user logged in from. 2 last hosts and dates are remembered. <b>Nullable</b>
☰	host2	nvarchar(50)	Name of second host that user logged in from. 2 last hosts and dates are remembered. <b>Nullable</b>
☰	host1_last_login	datetime	Date and time of last login from first host <b>Nullable</b>
☰	host2_last_login	datetime	Date and time of last login from second host <b>Nullable</b>
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL

### Linked from

	Table	Join	Title / Name / Description
→	dbo.user_connections (User connection details)	dbo.licenses.license_id = dbo.user_connections.license_id	FK_user_connections_licenses

## Unique keys

	Columns	Name / Description
🔑	license_id	PK_licenses Table primary key

	Columns	Name / Description
🔑	login	UK_licenses_login Login is unique in repository user database

## Triggers

	Name	When	Description
⚡	trg_licenses_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.8.1.4. Table: dbo.version (Repository version)

Holds information about repository version and updates. Each update to a new version creates a new row in table. If Version.update is greater than used Dataedo application version, you won't be able to connect to it.

### Columns

	Name	Data type	Description / Attributes
☰	version_entry_id	int	Row id <b>Identity / Auto increment</b>
☰	version	int	Dataedo version (eg. 4)
☰	update	int	Repository update. This is still the same version, but a patch has been applied. Eg. 4.1 <b>Nullable</b> <b>Default:</b> 0
☰	release	int	Not used anymore <b>Nullable</b> <b>Default:</b> 0
☰	stable	bit	1 when installation/upgrade was performed sucessfully, 0 when it was stopped for some reason. <b>Nullable</b>
☰	installation_date	datetime	Time of repository installation/upgrade. <b>Nullable</b> <b>Default:</b> getdate()
☰	installed_by	nvarchar(100)	Name of Database/Dataedo user that upgraded repository. <b>Nullable</b> <b>Default:</b> NULL
☰	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
☰	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
☰	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
☰	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL

### Unique keys

	Columns	Name / Description
🔑	version_entry_id	PK_version Table primary key
🔑	version, update, release	UK_version Version and update must be unique

## Triggers

	Name	When	Description
	trg_version_Modify	After Insert, Update	Standard insert/update trigger setting last_modification_date, modified by columns

## 2.9. Other

### 2.9.1. Tables

#### 2.9.1.1. Table: dbo.custom\_field\_classes (Field class)

Dataedo Web will present custom fields grouped into following classes:

1. Description - together with native description field
2. Label - for fields such as Form label
3. Status - for fields that specify status of the data element
4. Role - for fields such as Owner, Steward, Architect...
5. Classification - such as Sensitivity Classification, GDPR Classification. Created by Data Classification module and can be created manually/
6. Domain - Created by Data Classification module, to indicate domain/type of the data, such as - email, name, address.
7. Format - for fields specifying format of the data
8. Mapping - for mapping fields, such as Source table/column
9. Other - for any other fields

This table holds this list.

#### Columns

	Name	Data type	Description / Attributes
PK	custom_field_class_id	int	Row id <b>Identity / Auto increment</b>
PK	code	nvarchar(100)	System-friendly unique text identifier
PK	name	nvarchar(100)	Human-friendly field class name
PK	sort	int	Default sort order
PK	creation_date	datetime	Standard row creation timestamp <b>Default:</b> getdate()
PK	created_by	nvarchar(1024)	Dataedo username that created row <b>Nullable</b> <b>Default:</b> NULL
PK	last_modification_date	datetime	Standard row last update timestamp <b>Default:</b> getdate()
PK	modified_by	nvarchar(1024)	Dataedo username that last updated row <b>Nullable</b> <b>Default:</b> NULL
PK	source_id	int	For internal use only <b>Nullable</b>

#### Linked from

	Table	Join	Title / Name / Description
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.classifiers.custom_field_1_class_id	FK_classifiers_custom_fields_classes_1
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.classifiers.custom_field_2_class_id	FK_classifiers_custom_fields_classes_2

	Table	Join	Title / Name / Description
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.classifiers.custom_field_3_class_id	FK_classifiers_custom_fields_classes_3
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.classifiers.custom_field_4_class_id	FK_classifiers_custom_fields_classes_4
→	dbo.classifiers (Data Classification domain)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.classifiers.custom_field_5_class_id	FK_classifiers_custom_fields_classes_5
→	dbo.custom_fields (Custom fields)	<b>dbo.custom_field_classes</b> .custom_field_class_id = dbo.custom_fields.custom_field_class_id	FK_custom_fields_custom_field_classes

### Unique keys

	Columns	Name / Description
🔑	custom_field_class_id	PK_custom_field_classes