Technical Forums (/s/forums) Knowledge Base (/s/knowledge-base) Ideas (/s/bridea/acideasULT__bı Log in Q0(

Article

← FME Desktop (/S/Topic/0TO4Q000000QL9uWAG/Fme-…

# Introduction to Working with Geodatabase Relationship Classes

Jun 9, 2022 • Knowledge

**Product Type**
FME Desktop

**FME Version**
2022.0

**Tutorial:** Tutorial: Geodatabase Transformations (/s/article/tutorial-getting-started-with-complex-geodatabase) **| Previous:** Working with Geodatabase Feature-Linked Annotation (/s/article/working-with-feature-linked-annotation) **| Next:** Writing Geodatabase Relationship Classes (/s/article/writing-geodatabase-relationship-classes)

## Introduction

Geodatabase relationship classes are used to manage the relationships between features in one feature class with features in another. Attributed and non-attributed relationship classes can be read and written with FME. Relationships are not rows in a table or feature class like other features, but linkages through the primary and foreign key values of an origin and destination feature.

“Origin” features belong to the “Origin Table/Feature Class” specified when creating the relationship class in ArcCatalog, while “destination” features belong to the “Destination Table/Feature Class”.

Relationship classes can be simple or composite, as well as attributed or non-attributed.

For more information regarding relationships in ArcGIS, please refer to Relationships and ArcGIS Esri Documentation (https://desktop.arcgis.com/en/arcmap/10.7/manage-data/relationships/relationships-and-arcgis.htm).

*Note that the Esri File Geodatabase reader/writer used/mentioned in this article requires a licensed version of ArcGIS. For more information on required ArcGIS license levels, please see* Required ArcGIS License Types for FME Geodatabase Formats. (https://community.safe.com/s/article/required-arcgis-license-types-for-fme-geodatabase) *The Comparison of FME readers and writers for SDE and Geodatabase (/s/article/comparison-of-fme-readers-and-writers-for-sde-and) may also be useful.*

**Relationship examples are available in** Writing Geodatabase Relationship Classes (/s/article/writing-geodatabase-relationship-classes)

### Free Trial

Download a free 30-day trial of FME Desktop (https://www.safe.com/fme/trial/) and start integrating your workflows. The free trial comes with the appropriate licensing to utilize all of the Esri functionality inside of FME.

## Video

## Content Overview

## Relationship Classes

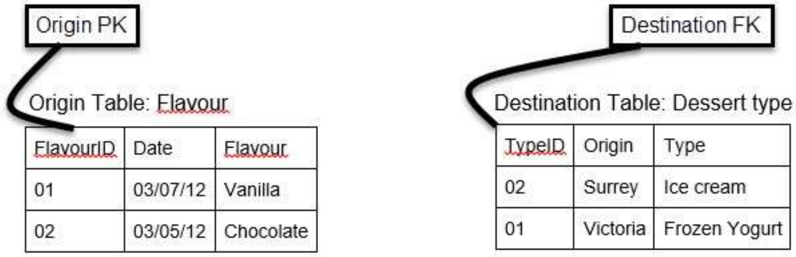
### Simple vs. Composite:

In a “Simple” Relationship class, features can exist independently of each other – for example, a feature can exist without necessarily having an attachment, and any attachment can be deleted without affecting the feature.

In a “Composite” Relationship class, origin and destination objects are more closely connected. If a feature were deleted so too would be the related attachment record.

### Attributed vs. Non-attributed:

In a non-attributed relationship, the relationship class is contained in a non-spatial table that only contains the origin and destination feature keys.



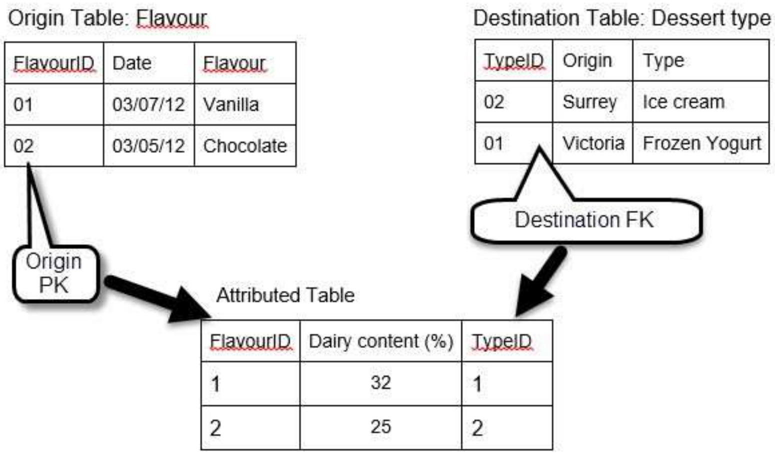
Origin Table: Flavour

| FlavourID | Date | Flavour |
|---|---|---|
| 01 | 03/07/12 | Vanilla |
| 02 | 03/05/12 | Chocolate |

Destination Table: Dessert type

| TypeID | Origin | Type |
|---|---|---|
| 02 | Surrey | Ice cream |
| 01 | Victoria | Frozen Yogurt |

In an attributed relationship, the relationship class is contained in a non-spatial table with more attributes than just the identifying keys.



Origin Table: Flavour

| FlavourID | Date | Flavour |
|---|---|---|
| 01 | 03/07/12 | Vanilla |
| 02 | 03/05/12 | Chocolate |

Destination Table: Dessert type

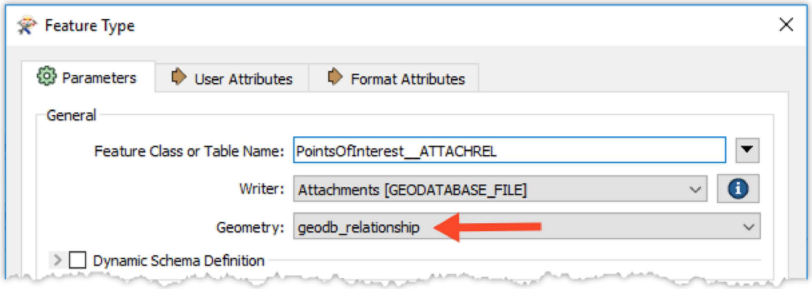
| TypeID | Origin | Type |
|---|---|---|
| 02 | Surrey | Ice cream |
| 01 | Victoria | Frozen Yogurt |

Attributed Table

| FlavourID | Dairy content (%) | TypeID |
|---|---|---|
| 1 | 32 | 1 |
| 2 | 25 | 2 |

## Reading Relationship Classes

When reading a relationship class, both the origin and destination feature classes must be read at the same time as the relationship class. A relationship feature type shows an allowed geometry of either *geodb_relationship* or *geodb_attributed_relationship.*



Feature Class or Table Name: PointsOfInterest__ATTACHREL

Writer: Attachments [GEODATABASE_FILE]

Geometry: geodb_relationship

☐ Dynamic Schema Definition

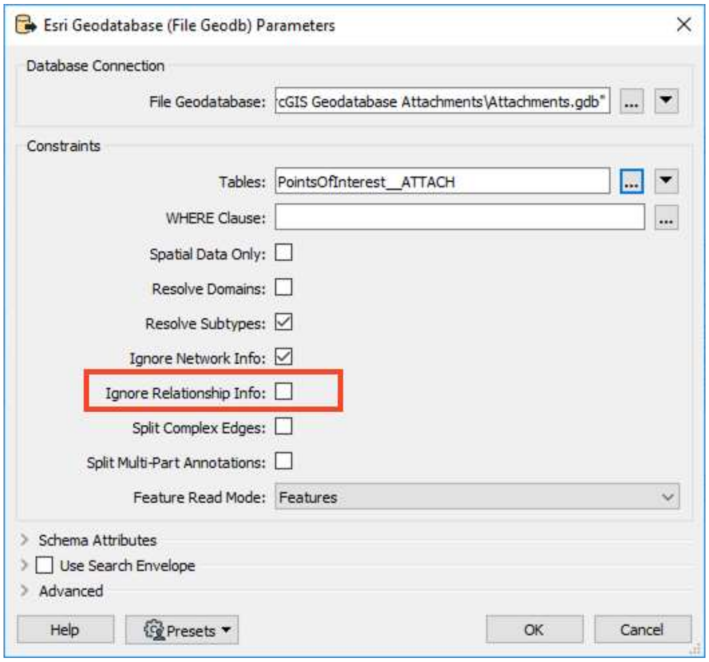
Each relationship feature has the following Format Attributes stored on it when read from a relationship class:

- *geodb_rel_origin_oid* - the primary key of the related origin feature which was used when the relationship was set up using ArcGIS.
- *geodb_rel_destination_oid* - the primary key of the related destination feature which was used when the relationship was set up using ArcGIS.

These attributes can be used to identify relationships between the origin and destination features within the workspace. These attributes are used for FME's processes, as format attributes, and do not appear in the final written output.

### Performance

Note that reading from relationship classes is very slow since each relationship is validated when read. Because this functionality is not often required, FME includes a parameter that, by default, turns off relationship reading to improve performance. Therefore, to read relationship classes you must first locate and deactivate the "Ignore Relationship Info" parameter.



File Geodatabase: cGIS Geodatabase Attachments\Attachments.gdb"
Tables: PointsOfInterest__ATTACH
WHERE Clause:
Spatial Data Only: □
Resolve Domains: □
Resolve Subtypes: ☑
Ignore Network Info: ☑
Ignore Relationship Info: □
Split Complex Edges: □
Split Multi-Part Annotations: □
Feature Read Mode: Features
□ Use Search Envelope

## Writing Relationship Classes

Relationship classes cannot be created through FME and must be set up through ArcGIS prior to running the translation. Once the relationship has been defined in ArcGIS, the following attributes must be stored on features written to a relationship class table in FME (**this only relates to M:N and attributed relationships because you interact with the relationship table directly**):

- *geodb_rel_origin_oid* - the ID attribute to be used from the related origin feature
- *geodb_rel_destination_oid* - the ID attribute to be used from the related destination feature
- *geodb_type* - either geodb_relationship or geodb_attributed_relationship

The following attribute must be stored on features written to the origin and destination feature classes/tables which participate in the relationship (this applies to all types of cardinality (1:1, 1:M, M:N and attributed)):

- *geodb_oid* - the ID attributes which were used in the geodb_rel_origin_oid and the geodb_rel_destination_oid.
- *geodb_feature_has_relationships* - Set to “YES” to specify that a feature participates in a relationship as an origin or destination.

**Note:**

- the _oid values mentioned above are not transferred to the Geodatabase. They are only supplied so that the Geodatabase writer knows which features are related.
- the _oid values must be **integers**

Required attributes for writing to relationship classes:

| Object | Required Attributes |
|---|---|
| Origin feature class or table | geodb_oid<br>geodb_feature_has_relationships = *yes* |
| Destination feature class or table | geodb_oid<br>geodb_feature_has_relationships = *yes* |

| | |
|---|---|
| Relationship class | geodb_rel_origin_oid<br>geodb_rel_destination_oid<br>geodb_type = *geodb_relationship* or *geodb_attributed_relationship*<br>Table Handling: **Use Existing** |

For example, if an origin feature has geodb_oid = 1 and a destination feature has geodb_oid = 2, the feature written to the relationship table must have these attributes:

geodb_rel_origin_oid = 1

geodb_rel_destination_oid = 2

## Relationships: insert, update or delete

Note that features participating in an **attributed relationship** can be inserted, updated and deleted. But features participating in a **non-attributed relationship** can only be inserted and deleted. Because attributed relationships have intermediate tables associated with them, the attributes of the intermediate table (excluding key values), may be updated by providing an RID (relationship id) as an update/key field.

Relationship Keys

In a geodatabase, the relationship keys do not have to be of type Object ID (integers). They could be a GUID for example. In this case, it's tempting to load the foreign key attributes in the relationship. You don't need to do this. You only need to set the values of geodb_rel_origin_oid & geodb_rel_destination_oid. The FME Geodb writer will populate the relationship feature class foreign keys.

**Use only one Geodatabase writer:** Be careful if you have more than one Geodatabase writer in your workspace. The origin and destination feature classes (or tables) that participate in the relationship and the relationship class must be written by the same Geodatabase writer (i.e. you cannot write to feature classes with one Geodatabase writer and to relationship classes with another).

## Inserting into a M:N or attributed relationship class table

Sometimes, the data in the origin and destination feature classes already exists and you just want to **add the relationships** to the relationship table. You can do this if:

- the relationship is M:N or an attributed relationship

- the relationship class already exists in your geodb

Set the geodb_type to geodb_relationship or geodb_attributed relationship (if attributed) and the geodb_rel_origin_oid and geodb_rel_destination_oid to their respective OBJECTID's. This does require that FME reads the feature classes which participate in the relationship but does not involve writing them out again. Only the relationship table needs to be written to in this case.

## Relationships Background

Relationships must be created using the ArcGIS tools, as they cannot be created in FME.

**1:1, 1:M non-attributed relationships:** These have no intermediate table for keeping track of relationships. They are linked using a primary (origin table) and foreign (destination table) key.

**M:N and any 1:1, 1:M or M:N attributed relationships:** Separate tables are used to keep track of the relationships. In order to track these relationships, each relationship entry contains a unique RID value along with the primary and foreign key.

**File Geodatabase and SDE Geodatabase:**

**1:1, 1:M non-attributed feature classes:** If inserting with FME you would either be starting with empty feature classes (because as soon as you create the relationship in ArcMap, the relationship is already completed), or you would be deleting and re-inserting features which participate in the relationship and thus rebuilding that relationship.

**File Geodatabase ONLY - not SDE geodatabase:**
**1:1, 1:M, M:N attributed relationships and M:N non-attributed relationships:** If you have populated feature classes and need to insert into the relationship table to populate it, or you want to update or insert attributes in an attributed relationship table, you may simply insert into the relationship table directly without writing to the feature classes which participate in the relationship. This includes the primary/foreign key and/or attributes. The assumption here is you already have a table or some source with those relationships defined. It's probably a less common workflow but it's doable and perhaps a good advanced post or just something we add to one of the articles as an aside. Generally, you will be inserting to both the feature classes and building/inserting into the relationship table at the same time.

**SDE Geodatabase:**
**1:1, 1:M, M:N attributed relationships and M:N non-attributed relationships:**

FME may only be used to UPDATE the attributes in an attributed relationship, not the RID, foreign and primary keys. If this is desirable, a DELETE and INSERT operation must be performed.

Currently, there isn’t a way to insert existing relationship attributes into a relationship table with a SDE Geodatabase without reading the features participating in the relationship in order to get the ObjectID. This is in part because populating these relationships in ArcGIS requires the origin and destinations to be picked interactively rather than just defining the relationship. As such, If the features which participate in the relationship classes already exist in the database and only the relation table needs to be populated, the features which participate must still be read. This is required in order to get the ObjectID’s, however these features do not be to be written back. The ObjectID's read from the origin and destination relationship features will need to be renamed to the geodb_rel_origin_oid and geodb_rel_destination_oid respectively in order for the relationship table to be populated.

# Deleting Records from a Relationship Class

A little more on deleting records for a relationship class...

You can delete individual records from a relationship class. To do this you can:

- delete the features from the participating feature classes. This will depend on the relationship (1:M, N:M etc.) But if you delete the participating features then the associated relationship record should be removed. For example, for a 1:M you should only have to remove the origin (parent) feature.
- delete the relationship record directly. To do this you need to set both the geodb_rel_origin_oid and geodb_rel_destination_oid. These act like a unique key to the relation record. You do need to set a key field on the FME Geodb writer (i.e. RID), but this is ignored for relationship classes.

## Continue to the Next Article: Writing Geodatabase Relationship Classes (https://community.safe.com/s/article/writing-geodatabase-relationship-classes)

**First Published Date**
7/29/2020, 12:17 AM

**Last Published Date**
6/9/2022, 10:27 PM

Transformation (/s/topic/0TO4Q000000...

FME Desktop (/s/topic/0TO4Q000000...

Sort by:
Latest Posts

geomartino (/s/profile/0054Q00000EwxRxQAJ)
March 17, 2017 at 3:49 PM (/s/feed/0D54Q000080iRVHSA2)

Hi all, thx for the tutorial.

I'm trying to run it, I keep receiving this error message:

Unable to create table/feature class/relationship class of type 'geodb_invalid'. Creation of this type is not currently supported by FME, and must be done through Esri tools

GeodatabaseRelationshipFeaturesPipeline::GeodatabaseRelationshipFeatures(SortFactory): A fatal error has occurred. Check the logfile above for details

Expand Post

5 comments 38 views

Like Comment

More comments 1 of 5

trentatsafe (/s/profile/005a000000CdnWnAAJ) (Employee)
5 years ago

Hi @geomartino,

You can share your questions or future questions over on the Q&A forum linked here:

Expand Post

Like

Log In to Comment

Follow

Files (1) (/s/relatedlist/ka14Q000000wn58QAA/AttachedContentDocuments)

ZIP attachments
Jul 28, 2020 • 484KB • zip

View All (/s/relatedlist/ka14Q000000wn58QAA/AttachedContentDocuments)

## Related Articles

Writing Geodatabase Relationship Classes (/s/article/writing-geodatabase-relationship-classes)

Working with Geodatabase Subtypes: Writing A Subtype (/s/article/working-with-geodatabase-subtypes)

Working with Geodatabase Domains: Writing A Coded Domain (/s/article/working-with-geodatabase-domains)

Working with Geodatabase Metadata: Writing/Updating Metadata (/s/article/working-with-geodatabase-metadata-writing-to-xml)

Working with Geodatabase Feature-Linked Annotation (/s/article/working-with-feature-linked-annotation)

SAFE SOFTWARE®

(https://safe.com)

Getting Started (../s/topic/0TO4Q000000QKioWAG/welcome/)

Ideas (../s/bridea/acideasULT brIdea c/00B4Q000000fWhEVUA1)

Feedback (https://forms.gle/LkoFWpziDYaWQkL78)

Forums (../s/forums/)

Groups (../s/group/CollaborationGroup/00Ba000000A0BxJEAV)

Knowledge Base (../s/knowledge-base/)

Support (../s/support/)

Register / Log In (/s/login/)

**Land Acknowledgement —**

Safe Software respectfully acknowledges that we live, learn and work on the traditional and unceded territories of the Kwantlen, Katzie, and Semiahmoo First Nations.