관계 클래스 (Relationship Class) 개념 및 활용

제품 : ArcGIS Desktop

제작일 : 2018 년 9 월 4 일

제작 : 한국에스리 기술지원본부

esri Korea
THE SCIENCE OF WHERE™

개요

본 문서는 ArcGIS Desktop 에서 관계 클래스 (Relationship Class) 의 개념 및 기본적인 사용 방법을 정리한 한국에스리 기술문서입니다.

이 문서와 관련된 내용에 대한 문의/건의 등을 원하신다면, 다음의 연락망을 통하여 한국에스리 기술지원센터로 연락주시기 바랍니다.

- 한국에스리 기술지원센터 (유지보수 고객 대상)
  - 고객지원 홈페이지 : http://www.esrikr.com/self-service/
  - 이메일 : help@esrikr.com
  - 전화 : 080-850-0915 | 운영시간: 평일 오전 9 시 ~ 오후 6 시
- 24 시간 기술지원 리소스 :
  - 한국에스리 기술자료 : http://esrikr.com/article-categories/technical/
  - Esri 기술지원 페이지(영문) : http://support.esri.com
  - 소프트웨어 개선 요청(영문) : http://ideas.arcgis.com
- ArcGIS Pro 도움말 : http://pro.arcgis.com/en/pro-app/help/
- ArcMap 도움말 : http://desktop.arcgis.com/en/arcmap/
- 원문 링크:

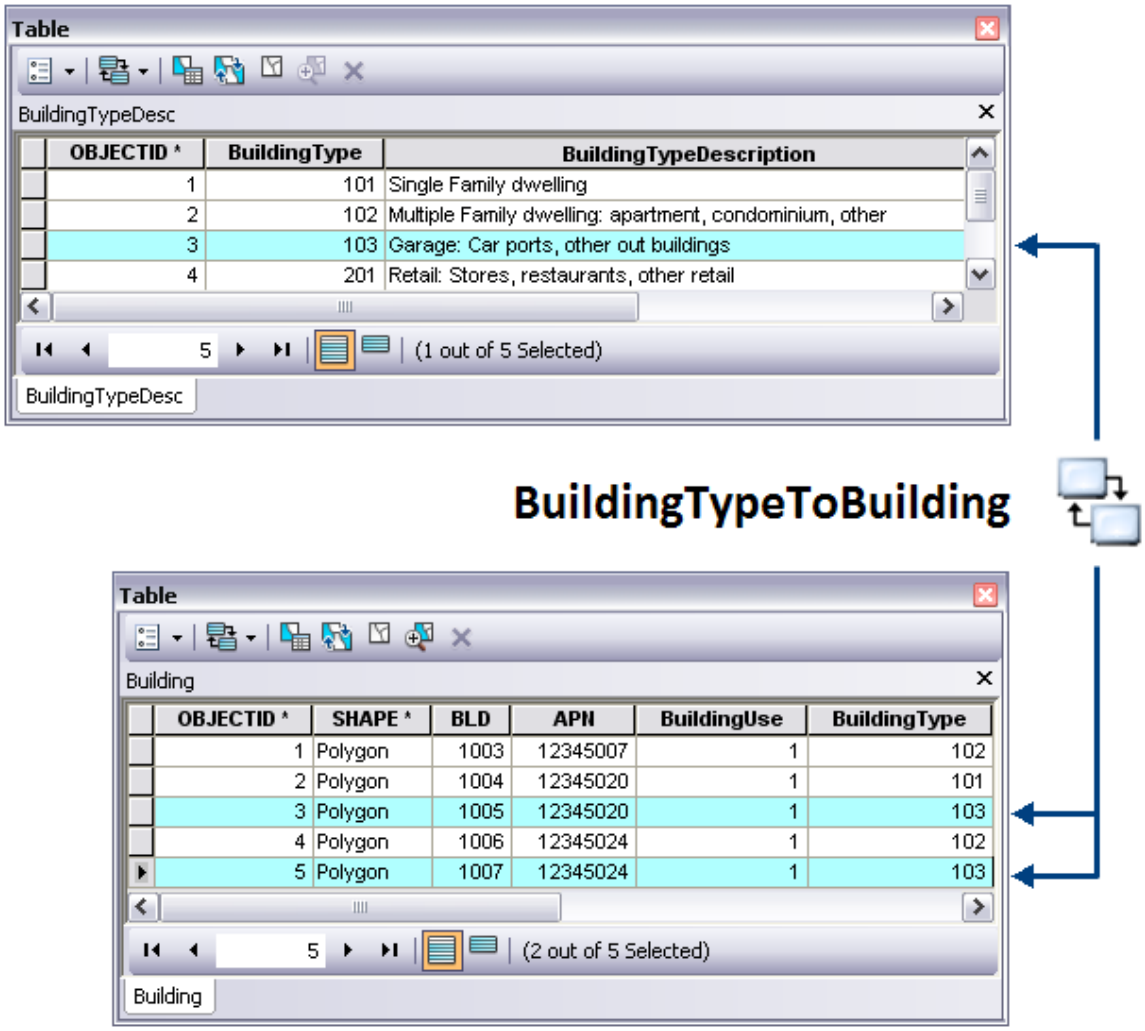
  http://desktop.arcgis.com/en/arcmap/latest/manage-data/relationships/benefits-of-relationship-classes.htm

  http://desktop.arcgis.com/en/arcmap/latest/manage-data/relationships/relationship-class-properties.htm

Relationship class 의 장점

Standard, Advanced 라이선스에서 사용 가능합니다.

지오데이터베이스의 관계 클래스(Relationship classes)는 한 클래스 (피처클래스 또는 테이블)의 객체와 다른 클래스의 객체간의 연결을 관리합니다. 관계의 양쪽 끝의 객체는 지오메트리가 있는 피처이거나, 테이블의 레코드일 수 있습니다



관계 클래스는 모든 카디널리티(일대일, 일대다, 다대다)를 지원하며 관계 자체에 대한 특성을 가질 수 있습니다.

또한 관계 클래스는 ArcMap 의 조인(join), 릴레이트(relate)에서는 지원되지 않는 다양한 고급 기능을 제공합니다.

- 관계 클래스는 관련 객체간의 참조 무결성을 강화하는데 사용할 수 있습니다.

  관계 클래스는 객체를 수정할 때 관련된 객체가 자동으로 갱신되도록 설정할 수 있습니다. 이는 관련 피처를 물리적으로 이동하거나, 관련 객체를 삭제하거나, 속성을 업데이트하는 작업이 포함될 수 있습니다. 예를 들어, 전신주를 이동할 때마다 해당 전신주에 부착된 변압기 및 기타 장비가 함께 이동할 수 있도록 관계를 설정할 수 있습니다. 규칙을 설정함으로써 관계 클래스가 유효한 관계 유형을 제한할 수 있습니다. 예를 들어 하나의 기둥에는 변압기를 최대 3 개 부착할 수 있습니다. 철제 전신주는 A 등급 변압기를 지원할 수 있지만 B 등급 변압기는 지원하지 않을 수 있습니다. 관계 클래스는 관련 클래스간의 참조 무결성을 적극적으로 유지합니다. 그 중 하나가 ArcMap 세션에 추가되지 않은 경우에도 마찬가지입니다.

- 관계 클래스를 사용하면 편집 작업이 쉬워 유지관리 비용을 줄일 수 있습니다.

  관계 클래스는 관련 객체에 대한 자동 갱신을 제공하여 사용자가 추가적인 편집을 하지 않아도 됩니다. 관계 클래스를 사용하면 편집하는 동안 객체에 접근할 수 있습니다. 객체를 선택한 다음 속성 대화 상자나 표를 사용하여 모든 관련 객체를 찾을 수 있습니다. 일단 관련된 객체를 탐색했다면, 그것의 속성을 편집할 수 있습니다. 얼마나 복잡하게 연결 되어있는지에 관계없이 모든 관계 클래스는 편집할 수 있습니다. 관계 클래스는 지오 데이터베이스에 저장되므로 버전으로 관리 할 수 있습니다. 버전을 사용하면 여러 사용자가 동시에 관계 피처나 레코드를 편집 할 수 있습니다.

- 관계 클래스를 사용하여 관련된 피처와 레코드에 질의(query)를 할 수 있습니다. ArcMap 의 조인과 마찬가지로 관련 클래스의 속성을 사용하여 쿼리를 수행하고 분석하며 보고서를 생성할 수 도 있습니다.
- 다른 resolution 에 대한 제한은 없습니다. 단순, 복합 관계 클래스는 서로 다른 resolution 의 피처클래스로 구성될 수 있습니다.

연관된 주제

Relationship class 속성 정보

ArcMap 10.6

Other versions

Standard, Advanced 라이선스에서 사용 가능합니다.

노트:

ArcGIS Desktop Advanced 와 Standard 에서는 관계 클래스 (Relationship class) 생성 및 편집이 가능하지만 Basic Edition 에서는 읽기 전용이며 관계 클래스에 사용되는 피처 또한 읽기 권한만 있습니다.

관계 클래스 (Relationship class)에는 원본 객체가 대상 객체와 어떠한 연관이 있는지 정의하는 몇 가지 속성 정보가 있습니다. 관계 클래스 생성 시 속성을 선택할 수 있습니다.

- 유형: 단순 또는 복합 (Simple or composite)
- 원본와 대상 클래스 (Origin and destination classes)
- 기본 키와 외래 키 (Primary and foreign keys)
- 카디널리티 (Cardinality): 대응비가 일대일, 일대다, 다대다의 여부
- 메시지 알림 방향, 사용자 정의 cascade 업데이트 및 삭제를 구현하려는 경우 적용 가능.
- 각 관계에 대한 속성을 저장할지 여부
- 이름 (Name)
- ArcMap 에서 관련 레코드를 탐색할 때 표시되는 정방향, 역방향 레이블

관계 (Relationship)를 생성하면 규칙을 지정하여 카디널리티를 구체화할 수 있습니다.

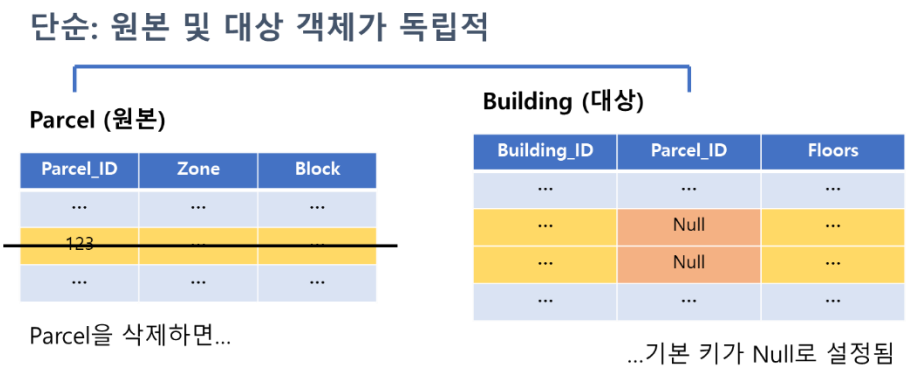
단순 vs 복합 (Simple vs Composite)

관계 클래스를 생성할 때 단순 또는 복합 관계를 지정할 수 있습니다.

단순 관계 (Simple Relationship)에서는 원본 객체는 서로 독립적으로 존재할 수 있습니다. 예를 들어 철도망의 경우, 철도 건널목과 하나 이상의 신호등이 연결될 수 있습니다. 그러나 신호등이 없는 철도 건널목이 존재할 수 있으며 철도 건널목이 없는 신호등이 철도망에 존재할 수 있습니다.

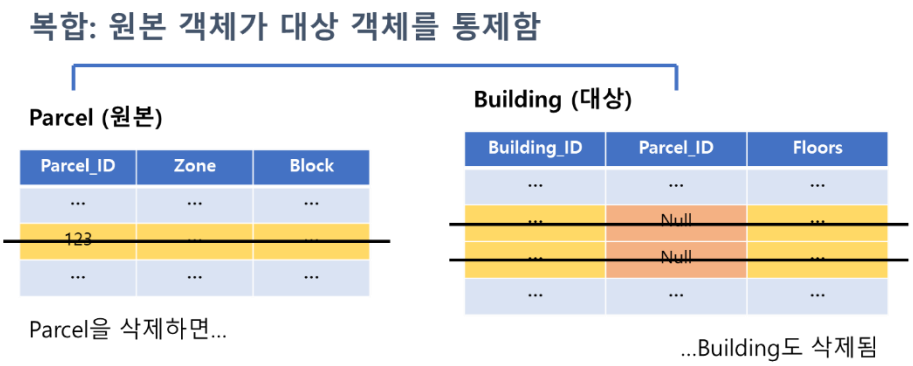
단순 관계에서 원본 객체를 제거하면, 일치하는 대상 객체에 대한 외래 키 필드 값이 Null 로 설정됩니다. 이 외래 키는 피처 간의 참조 무결성을 유지하도록 설계되었습니다. 원본 객체를 삭제하면 외래 키 값은 더 이상 해당 행을 원본 객체와 연결되어 있지 않으므로 외래 키 값은 더 이상 필요하지 않아 Null 로 설정됩니다. 외래 키의 유일한 목적은 대상 객체와 연결된 원본 객체 간의 관계를 유지하는 것입니다. 기본 키 값과 일치하는 원본 피처가 없으면 외래 키 값을 유지할 이유가 사라집니다. 나중에 하나의 대상 피처를 새로운 원본 피처 또는 다른 원본 피처에 연결시키기 위해 외래 키 필드를 Null 에서 새 외래 키 값으로 업데이트 할 수 있습니다.

대상 객체를 삭제해도 연결된 원본 객체의 기본 키 값에는 아무런 영향을 미치지 않습니다.



단순 관계의 대응비는 일대일, 일대다, 다대다 카디널리티를 가질 수 있습니다.

단순 관계처럼, 복합 관계 (Composite Relationship) 또한 객체를 삭제할 때 참조 무결성을 유지하지만 다른 방식으로 실행됩니다. 복합 관계에서 대상 객체는 원본 객체와 독립적으로 존재할 수 없으므로 원본이 삭제되면 연결된 대상 객체 또한 cascade delete 라는 과정을 통해 삭제됩니다.



이 종속 규칙은 참조 무결성을 테스트하기 위해 편집 세션에서 실행하는 명령인 'ArcMap 피처 유효성 (Validate Features)' 명령으로 적용됩니다. 대상 객체를 생성하고 원본 객체와 연결하지 않을 경우, 피처 유효성 (Validate Features) 기능은 오류가 나타납니다.

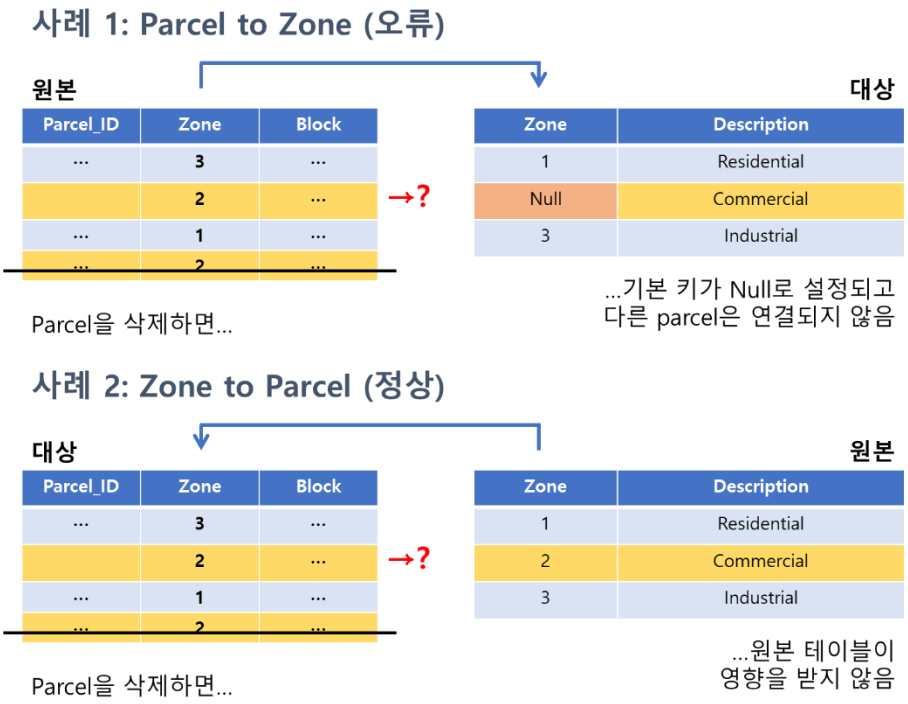
복합 관계는 피처 공간성을 유지하는데 도움이 됩니다. 메시징이 정방향 (Forward)으로 설정되어 있을 때, 원본 피처를 이동하거나 회전하면 연결된 대상 피처도 함께 이동하거나 회전합니다.

복합 관계는 사용자가 생성할 때 항상 일대다의 대응비로 구성되지만 경우에 따라 일대일 대응비로 제한할 수 있습니다.

원본과 대상 클래스 (Origin and Destination Classes)

관계 클래스를 생성할 때 하나의 클래스를 원본으로, 다른 클래스를 대상으로 선택할 수 있습니다. 두 가지를 구분 짓는 것은 매우 중요합니다. 복합 관계에서 cascade delete 기능으로, 이 점의 중요성을 확인할 수 있습니다.

단순 관계에서, 올바른 테이블을 확인하는 것은 중요합니다. 원본 클래스의 레코드를 삭제하면 대상 클래스에서 일치하는 레코드를 찾아 키 필드의 값을 Null 로 설정하기 때문입니다. 잘못된 클래스를 원본으로 선택하고 원본 객체를 삭제하면 외래 키 필드에 오류가 발생합니다. 예시는 아래 그림과 같습니다:



사례 1: Parcel to Zone (오류)

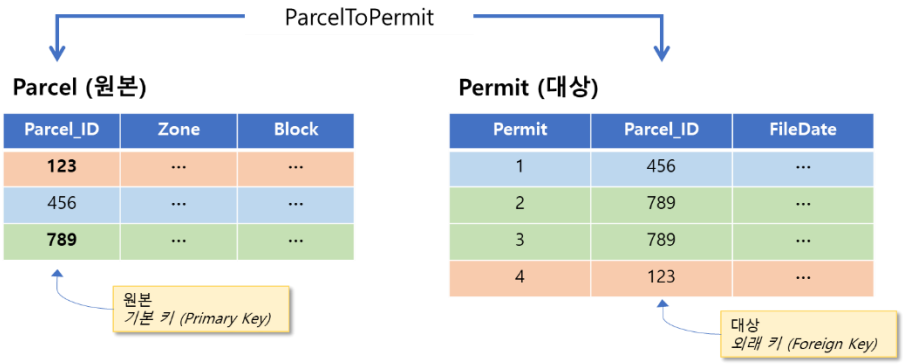
일반적으로 위와 같은 오류가 나타납니다. Zone 테이블에는 다른 Zone 코드에 대한 Description 이 있으며 개념적으로 ArcGIS Desktop Advanced Workstation 찾아보기 테이블과 유사합니다. 이 경우 Parcel 클래스가 원본이며 Zone 테이블이 대상입니다. 이것은 ArcGIS Desktop Advanced Workstation 에서 관계를 설정하는 방법입니다. 문제는 Parcel 을 삭제할 때 Zone 테이블의 일치하는 레코드에 대해 키 필드 (Zone)의 값이 Null 로 설정됩니다. 해당 Zone 코드가 있는 다른 Parcel 은 Zone 테이블과 일치하지 않습니다.

사례 2: Zone to Parcel (정상)

이 문제를 해결하려면 Zone 테이블을 원본으로 설정해야 합니다. Parcel (대상 객체)을 삭제해도 Zone 테이블에는 아무런 영향이 없으며 Zone 코드 (원본 객체)를 삭제하더라도 일치하는 Parcel 레코드의 Zone 필드 값을 Null 로 설정하면 됩니다. 이는 더 이상 일치하는 Zone 테이블 레코드가 없기 때문입니다.

기본 키 와 외래 키 (Primary and Foreign Key)

관계 클래스에서 원본 객체는 키 필드의 값을 통해 대상 객체와 일치시킵니다. 아래 예제에서, 모든 레코드가 동일한 Parcel ID 를 가지기 때문에 Parcel 789 는 Permits 의 2 와 3 에 대응합니다.



관계의 원본 클래스에 있는 키 필드를 기본 키라고하며 PK 로 축약됩니다. 실제 기본 키와 달리 관계의 기본 키 필드의 값은 모든 객체에 대해 고유해야 할 필요는 없습니다.

대상 클래스의 키 필드는 외래 키라고 하며 FK 로 축약됩니다. 원본 클래스의 기본 키 필드와 일치하는 값을 포함합니다. 키 필드 값은 각 행마다 고유해야 할 필요는 없습니다.

키 필드는 이름이 다를 수 있지만 동일한 데이터 유형이어야하며 Parcel ID 와 같은 종류의 정보를 포함해야 합니다. BLOB (Binary Large Object), 날짜 및 래스터를 제외한 모든 데이터 유형의 필드는 키 필드가 될 수 있습니다. 관계 클래스를 생성할 때 키 필드를 지정해야 합니다.

기본 키 필드를 결정할 때 한 가지 옵션은 일반적으로 ObjectID 필드라고하는 행 ID 필드를 사용하는 것입니다. ObjectID 필드는 피처 클래스 또는 테이블을 생성하거나 엔터프라이즈 지오데이터베이스에 레이어 또는 테이블을 등록 할 때 ArcGIS 에 의해 자동으로 추가됩니다. 이 필드는 각 레코드에 대해 고유 한 ID 를 보장합니다. ObjectID 는 ArcGIS 에 의해 유지 관리되므로 사용자가 수정할 수 없습니다.

지정된 객체의 ObjectID 값은 원본 클래스가 남아있는 한 변경되지 않으며 객체가 피처인 경우 피처를 분할하면 안됩니다. 피처를 분할하면 원래 피처의 지오메트리는 변경되지만 ObjectID 는 유지되며, 분할된 영역에는 새로운 ObjectID 가 부여된 피처가 생성됩니다. 결과적으로 원래 ObjectID 가있는 피처만 해당 ObjectID 값에 종속된 관계를 유지하게 됩니다.

이 때문에 ObjectID 필드를 사용하는 대신 사용자 지정의 기본 키 필드를 만들고 사용하는 것이 좋습니다. 다음은 각 조작을 수행 할 때 기본 키 필드가 어떻게 관계를 유지하도록 하는지 설명합니다.

- 레코드를 다른 피처 클래스 또는 테이블로 가져오면 새 ObjectID 값이 할당되고 원래 ObjectID 값을 기반의 관계가 손실됩니다. 대신 다른 기본 키에 관계를 설정하면 레코드가 가져올 때 기본 키의 ID 값이 변경되지 않습니다. 이렇게 하면 관련 클래스 세트를 새 클래스로 가져올 때 관계를 보존할 수 있습니다.

  예외는 복사 / 붙여 넣기 기능을 사용할 때입니다. 복사 / 붙여 넣기는 ObjectID 값을 유지합니다. 이 방법으로만 객체를 이동하려는 경우 ObjectID 필드를 기본 키로 사용할 수 있습니다.

- ObjectID 필드가 기본 키 필드로 사용되는 관계 클래스를 복제하면 동기화 중에 추가 처리가 필요하므로 성능에 영향을 미칠 수 있습니다. 경우에 따라 예기치 않은 동작이 발생할 수도 있습니다. 복제에서 ObjectID 를 기본 키로 사용하는 방법에 대한 자세한 내용은 Replicating related data 를 참조하십시오.

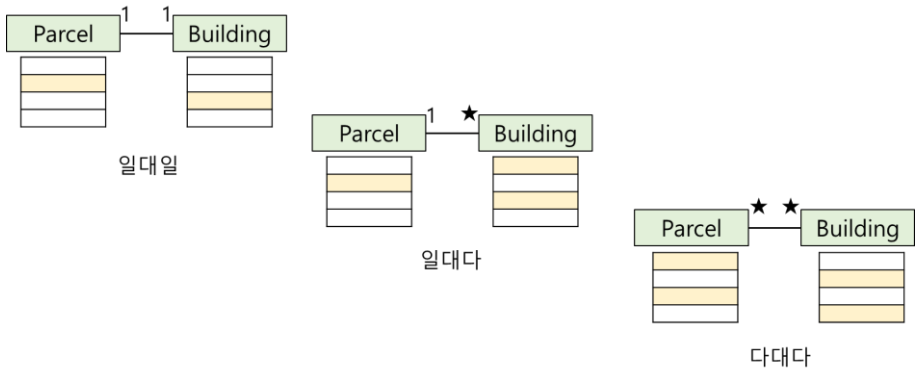
- 피처를 분할하면 원래 피처는 유지되며 (지오메트리는 업데이트 됨) 분할된 부분이 새로운 피처로 생성됩니다. 원본 ObjectID 를 기반으로 한 관계가 있는 경우 분할로 인해 생성된 두 가지 피처 중 하나만 관계만 유지됩니다. ObjectID 가 아닌 다른 필드를 키로 사용하면 피처를 분리할 때 원본 피처의 키 값이 두 개의 새 피처로 복사됩니다. 결과적으로 관련된 대상 테이블의 레코드가 두 가지 새로운 피처와 관계를 맺게 됩니다. 관계 클래스가 다대다로 설정된 경우 이상적입니다.

  피처를 분할하지 않고 모든 객체가 원본 클래스에 남아있을 것이라면 ObjectID 를 ID 로 사용할 수 있습니다. 이것을 보장 할 수 없다면, ObjectID 필드 대신 다른 고유 ID 필드를 설정하고 사용하는 것이 가장 좋습니다.

- 두 피처를 병합 할 때 새 피처는 원래 피처 중 하나의 ObjectID를 유지합니다. 피처를 병합하지만 피처를 클래스 밖으로 이동하거나 분할하지 않으려는 경우 ObjectID 필드를 기본 키로 사용할 수 있습니다.

카디널리티 (Cardinality)

관계의 카디널리티는 대상 클래스의 객체들과 관련될 수 있는 원본 클래스의 객체 수를 지정합니다. 관계는 세 가지 카디널리티 중 하나에 적용됩니다. :



일대일: 하나의 원본 객체는 하나의 대상 객체에만 관련될 수 있습니다. 예를 들어, Parcel 에는 Legal Description 이 하나만 있을 수 있습니다. ArcGIS 에서의 카디널리티는

다대일 관계를 다룰 수 있습니다. 다대일 관계의 예는 동일한 Legal Description 과 관련된 다수의 Parcel 입니다.

일대다: 하나의 원본 객체는 여러 대상 객체와 관련될 수 있습니다. 예를 들어, 하나의 Parcel 에 많은 Building 이 있을 수 있습니다. 일대다 관계에서는 일이 원본 클래스여야하고 다가 대상 클래스이어야 합니다.

다대다: 하나의 원본 객체는 여러 대상 객체와 관련될 수 있으며 반대로 하나의 대상 객체는 여러 원본 객체와 관련될 수 있습니다. 예를 들어, 주어진 property 는 많은 owner 를 가질 수 있고 주어진 owner 는 많은 property 를 소유 할 수 있습니다.

일(one)과 다(many)라는 용어는 오해의 소지가 있습니다. 일(One)은 실제로 영대일 이고 많은 것은 실제로 영대다 입니다. 예를 들어 Parcel 와 Building 사이에 일대다 관계를 만들 때 관계는 다음을 모두 허용합니다.

- Building 이 없는 하나의 Parcel
- Parcel 이 없는 하나의 Building
- Building 이 많은 하나의 Parcel

관계를 생성하면 규칙을 지정하여 카디널리티를 구체화할 수 있습니다. 대상의 여러 객체와 관련이 있는 원본 객체 수를 지정하는 규칙을 설정할 수 있습니다

관계 규칙 (Relationship rules)

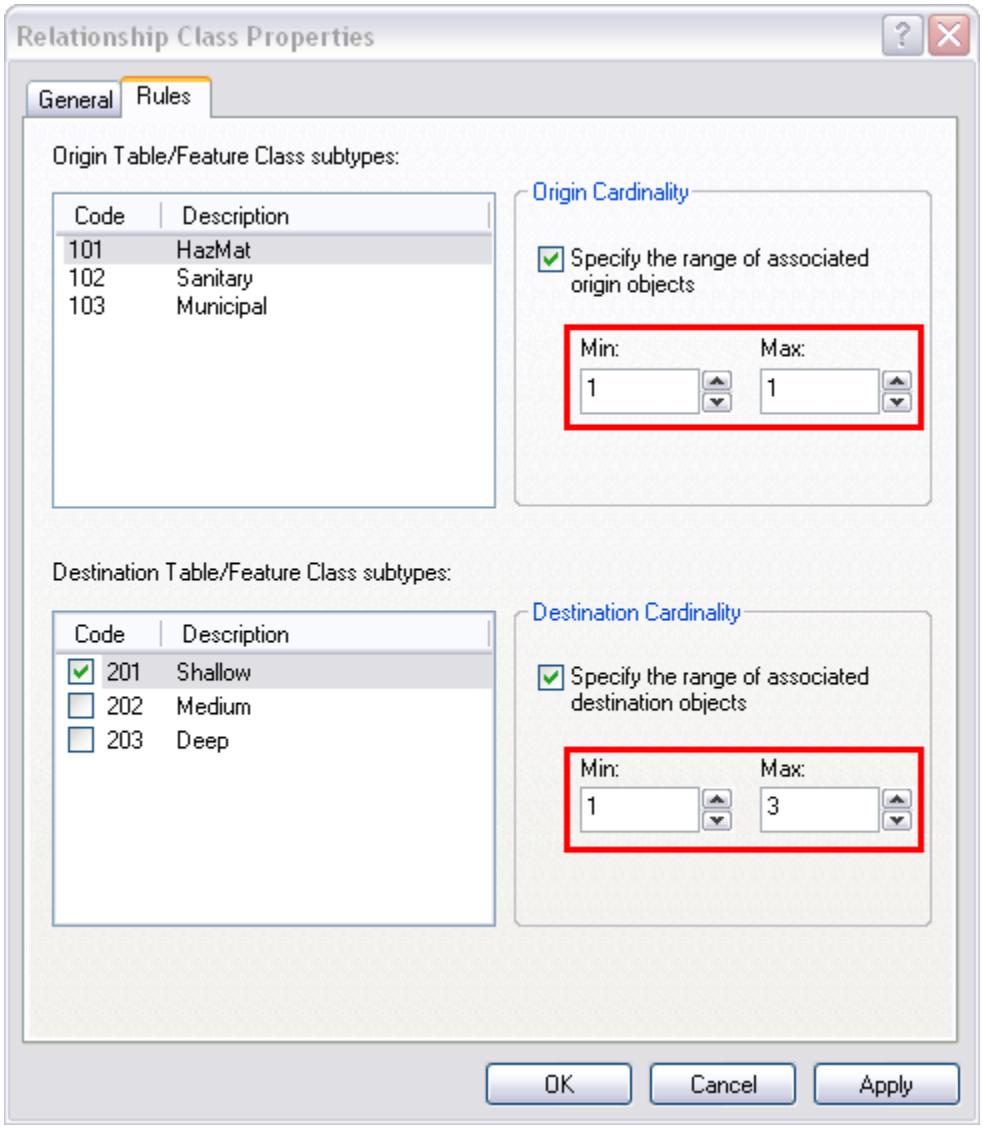
관계 클래스를 생성할 때, 일대일, 일대다, 다대다 의 카디널리티를 지정합니다.

관계는 종종 보다 제한적인 용어로 정의되어야 합니다. 예를 들어 Parcel 과 Building 의 관계에서 각 Building 을 Parcel 과 연관 시키거나 Parcel 에 최대 개수의 Building 을 포함시킬 것을 요구해야 할 수도 있습니다. 사용자가 Building 을 Parcel 에 연결하거나 너무 많은 Building 을 Parcel 에 연관시키는 것을 잊지 않도록 합니다.

하위 유형이 있는 경우 대상의 특정 유형의 객체와 관련된 원본 객체의 수와 유형을 제한할 수 있습니다. 예를 들어 철제 전신주는 A 변압기를 지원하지만 나무 전신주는 B 변압기를 지원합니다. 또한 각 유효한 하위 유형 쌍에 대해 허용되는 카디널리티 범위를 지정해야 할 수도 있습니다. 예를 들어 철제 전신주는 0-3 A 변압기를 지원할 수 있지만 나무 전신주는 0-2 B 변압기를 지원할 수 있습니다.

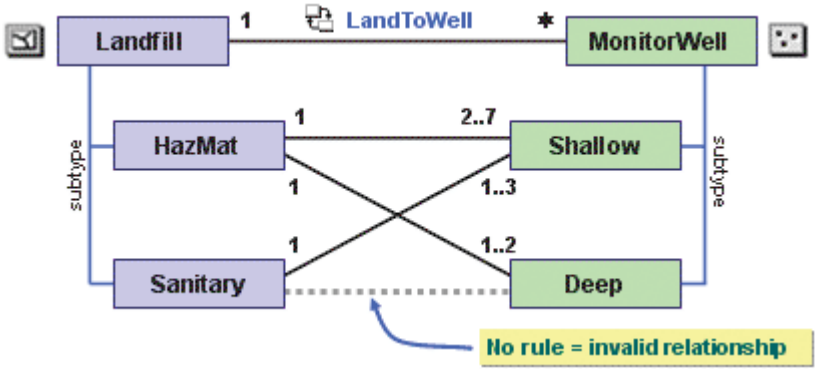
관계 클래스를 생성한 후에는 이러한 참조 무결성 규칙을 적용하는데 도움이 되는 규칙을 지정할 수 있습니다:

1. ArcCatalog 또는 Catalog 창에서 기존 관계 클래스를 마우스 오른쪽 버튼으로 클릭하여 관계 클래스의 Properties 대화상자를 열고 규칙 (Rules) 탭을 클릭합니다.
2. 원본 클래스에서 하위 유형을 선택하고 대상 클래스에서 대응하는 하위 유형을 선택합니다.
3. 원본 및 대상 카디널리티의 체크박스를 체크합니다. 규칙에 적합한 최소 및 최대 카디널리티를 설정하십시오. 아래의 대화상자는 Max 보다 큰 Min 을 설정하지 못하도록 하므로 Max 카디널리티 값을 먼저 설정합니다.



일단 관계 규칙이 관계 클래스에 추가되면, 그 규칙은 존재할 수 있는 유일한 유효한 관계가 됩니다. 다른 관계 조합과 카디널리티를 유효하게 하려면 관계 규칙을 추가해야 합니다.

아래의 예에서, HazMat 매립지는 1 개 또는 2 개의 깊은 우물 또는 2 개에서 7 개의 얕은 우물과 관련 될 수 있습니다. 그러나 위생 매립장 (Sanitary)이 깊은 우물과 관련이 있지만, 이 두 하위 유형 간의 규칙이 작성되지 않았다면 피처 유효성은 관계가 유효하지 않다고 간주합니다.



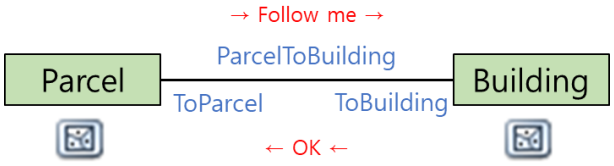
규칙을 설정하고 편집을 시작한 후 'ArcMap 피처 유효성 (Validate Features)' 명령을 사용하여 규칙을 테스트 할 수 있습니다. 피처 유효성 명령은 현재 선택된 피처가 관계 규칙을 위반하는지 여부를 알려줍니다.

메시지 알림 방향 (Message notification direction)

이전에 언급했듯이, 복합 관계에서 원본 객체를 삭제하면 연결된 대상 객체가 자동으로 삭제됩니다.

단순 관계 또는 복합 관계로 작업하는 경우, 원본을 업데이트하면 연결된 대상 피처 또한 업데이트 되어야 합니다. 또한 한 방향과 다른 방향, 또는 양방향 모두에서 업데이트 할 필요가 있습니다.

- 피처를 이동하거나 회전할 때 연결된 피처가 함께 이동하거나 회전하도록 함.
- 피처를 업데이트할 때 연결된 피처의 속성이 자동으로 업데이트 되도록 함.
- 원본 객체 업데이트는 연결된 대상 객체의 업데이트를 요구함.
- 대상 객체 업데이트는 연결된 원본 객체의 업데이트를 요구함.



관계에 이 동작이 필요한 경우, 원본 및 대상 객체가 변경될 때 알림 메시지를 보내 연결된 객체를 적절하게 업데이트 할 수 있습니다.

이 작업을 수행하려면, 관계 생성 시 메시지 알림 방향을 설정해야 합니다. 원본을 업데이트 하여 연결된 대상 객체의 업데이트가 필요한 경우 메시지 알림 방향을 정방향 (Forward)으로 설정합니다. 대상을 업데이트 하여 연결된 원본 객체의 업데이트가 필요한 경우 메시지 알림 방향을 역방향(Backward)으로 설정합니다. 위 두 가지가 모두 필요한 경우 메시지 알림 방향을 모두(Both)로 설정합니다. 일단 관계를 생성하면 메시지를 받는 객체에 동작을 프로그래밍하여 응답할 수 있도록 해야합니다.

예외. 복합 관계의 경우 메시지 방향이 정방향 (Forward)으로 설정된 경우

정방향 메시징을 사용하여 복합 관계를 생성하면 원본 객체를 이동하거나 회전시키면 연결된 대상 피처가 자동으로 이동하거나 함께 회전합니다. 관계를 올바르게 설정했다면 이 기능은 사용자 정의 프로그래밍이 필요 없습니다.

다른 메시지 알림 방향의 경우 사용자 정의 프로그래밍이 필요합니다. 정방향 메시지로 복합 관계를 생성하거나 사용자 지정 동작을 프로그래밍하려는 경우가 아니라면 메시지 알림을 없음으로 설정합니다. 그렇지 않으면 편집 작업을 수행할 때마다 메시지가 불필요하게 생성되어 기능이 저하됩니다.

복합 관계의 방향을 설정할 때 복합 관계에 있는 원본 객체가 삭제되면 대상의 모든 관련 객체가 자동으로 삭제되는 점에 유의하시기 바랍니다. 메시지 방향이 정방향, 역방향, 모두 또는 없음으로 설정된 여부에 관계 없이 연결된 객체는 삭제됩니다.

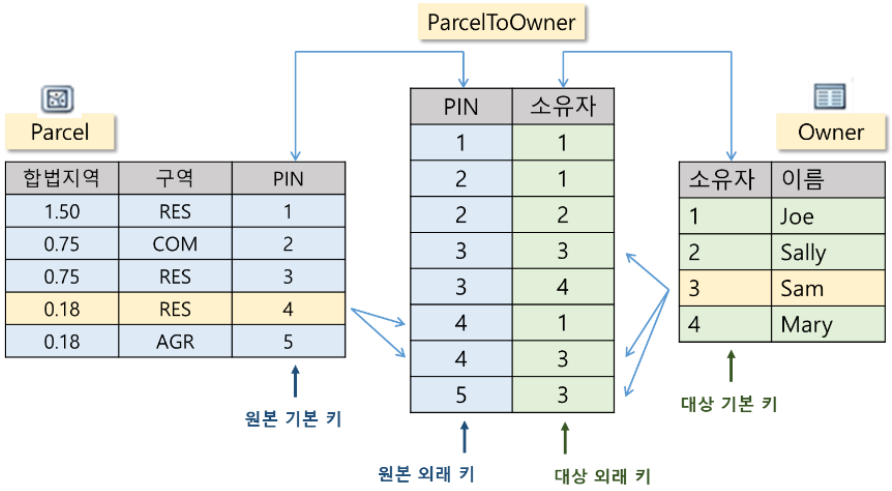
방향 (Direction)	**단순 관계에서의 영향** **(Effect on simple relationships)**	**복합 관계에서의 영향** **(Effect on composite relationships)**
정방향 (Forward)	사용자 정의 프로그래밍 설정하지 않으면 영향 없음	✓ 원본 삭제 시 대상 삭제 ✓ 원본 이동 및 회전 시 대상 이동 및 회전 ✓ 사용자 정의 프로그래밍 설정하지 않으면 다른 영향 없음

역방향 (Backward)	사용자 정의 프로그래밍 설정하지 않으면 영향 없음	✓ 원본 삭제 시 대상 삭제 ✓ 사용자 정의 프로그래밍 설정하지 않으면 다른 영향 없음
양방향 (Both)	사용자 정의 프로그래밍 설정하지 않으면 영향 없음	✓ 원본 삭제 시 대상 삭제 ✓ 원본 이동 및 회전 시 대상 이동 및 회전 ✓ 사용자 정의 프로그래밍 설정하지 않으면 다른 영향 없음
없음 (None)	메시지 전송 방지. 성능 향상	✓ 원본 삭제 시 대상 삭제 ✓ 이외의 메시지 전송 방지. 성능 향상

다대다 관계 (Many-to-many relationships)

일대일 및 일대다 관계에서 원본 클래스의 기본 키에 있는 값은 대상 클래스의 외래 키에 있는 값과 직접 연결됩니다.

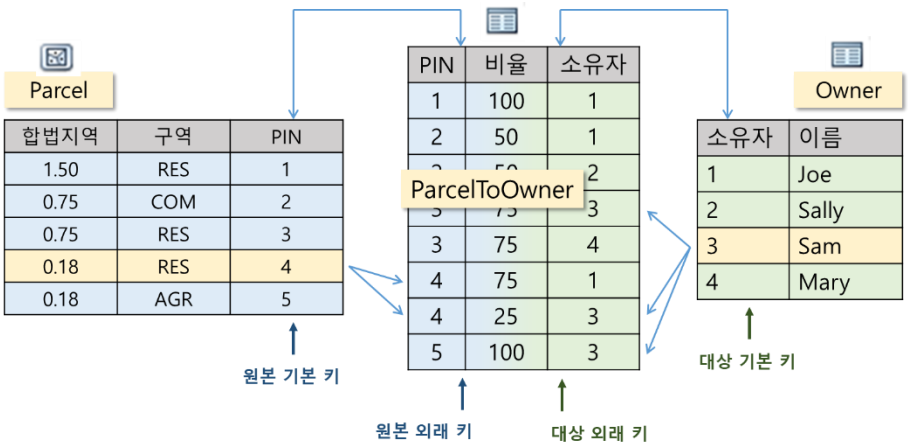
반면 다대다 관계는 연결을 매핑하기 위해 중간 테이블을 사용해야 합니다. 결과적으로 다대다 관계를 생성할 때 중간 테이블이 자동으로 생성됩니다. 중간 테이블은 원본 테이블의 기본 키 값을 대상 테이블의 외래 캐 값으로 매핑합니다. 각 행은 하나의 원본 객체를 하나의 대상 객체와 연결시킵니다.



중간 테이블을 생성하게 되면 필드만 생성됩니다. ArcGIS 는 어떤 원본 객체가 어떤 대상 객체와 연결되어 있는지 알지 못하므로 테이블에 수동으로 행을 생성해야 합니다. 이 테이블을 구성하는 것은 관계를 설정하는 데 가장 많은 시간이 소요되는 부분입니다.

관계 속성 (Relationship attributes)

다대다 관계의 중간 테이블은 선택적으로 관계 자체의 속성을 저장하는 용도로 사용할 수 있습니다. 예를 들어 Parcel 데이터베이스에서 Parcel 과 소유주 간의 관계 클래스가 있습니다. 소유자는 Parcel 을 소유하고 Parcel 은 소유자에게 소유됩니다. 각 관계의 속성은 소유권의 비율입니다. 이러한 속성을 저장해야하는 경우, 관계를 생성할 때 중간 테이블에 추가할 수 있습니다.



비록 유용하지 않지만, 일대일 또는 일대다 관계를 설정할 때 관계 속성을 저장할 필요가 있습니다. 이 경우 중간 관계를 생성할 때 테이블이 생성되도록 설정해야 합니다. 다대다 관계에서 중간 테이블은 원본의 기본 키 값을 대상의 외래 키 값에 연결하므로 각 관계에 대해 원하는 개수의 속성을 저장할 수 있습니다.

중간 테이블을 ArcCatalog 또는 Catalog 창에서 미리보기 할 수 있습니다. ArcMap 에 관계 클래스를 추가하면 중간 테이블을 열고 조작할 수 있습니다. ArcGIS 는 다른 작업을 위해 중간 테이블을 노출하지 않습니다. 예를 들어 ArcCatalog 또는 Catalog 창에서 속성을 표시하여 필드를 추가하거나 삭제할 수 없으며 기본 값이나 도메인의 사용을 지원하지 않습니다.

이름 (Name)

각 관계 클래스에는 카탈로그 트리 구조에 표시되는 이름이 있습니다. 데이터베이스 구조를 이해하기 쉽게 만들려면, 관계 클래스를 설명하는 이름을 지정하십시오.

원본 피처 클래스의 이름을 입력하고 Has 또는 Have 를 사용하여 대상 피처 클래스 이름으로 마무리합니다.

예) AddressHasZones 또는 ParcelsHaveOwners

원본 피처 클래스 이름이 다대일 또는 다대다 카디널리티인 경우 복수화하고 일대다 또는 다대다 카디널리티인 경우 대상 피처 클래스 이름을 복수화합니다.

이 방법을 사용하면 관계 클래스의 카디널리티를 이름에서 판별할 수 있습니다. 예를 들어 복수형의 두 피처 클래스의 이름 (ParcelsHaveOwners)을 보고 다대다 관계임을 확인할 수 있습니다.

정방향과 역방향 레이블 (Forward and backward labels)

정방향과 역방향 레이블은 속성에 표시되며 ArcMap 의 결과 대화 상자를 식별합니다. 또한 연결된 객체를 탐색하는 데 도움을 줍니다.

관계 클래스는 두 가지 레이블을 가집니다:

- 원본에서 대상으로 이동할 때 표시되는 정방향 레이블. 전신주 변압기 (Pole-transformer) 예시에서 이 레이블은 'supports'라고 읽을 수 있으며 전신주(pole)가 변압기를 지원 가능하다는 의미입니다.
- 대상에서 원본으로 이동할 때 표시되는 역방향 레이블. 전신주 변압기 (Pole-transformer) 예시에서 이 레이블은 'is mounted on'이라고 읽을 수 있으며 이 변압기가 전신주에 장착되어 있음을 의미합니다.

연관된 주제 (Related topics)

- An overview of creating and maintaining relationship classes
- Deciding between relationship classes, relates, and joins
- Modeling limitations of relationship classes
- Relationship classes: Maximizing performance