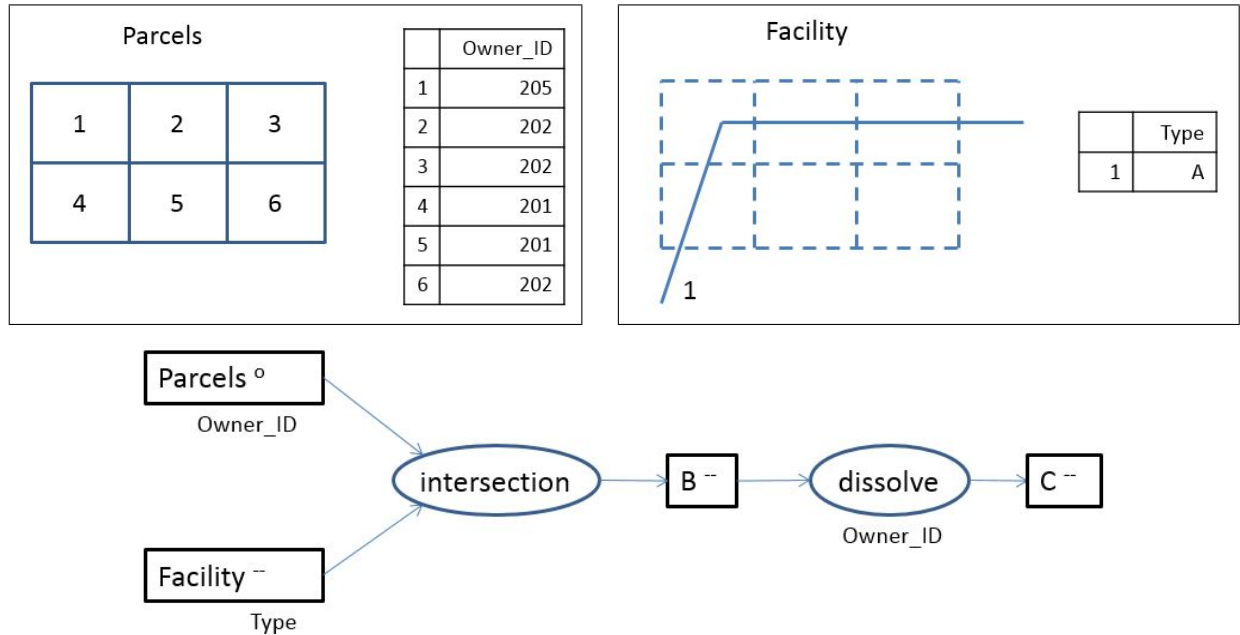
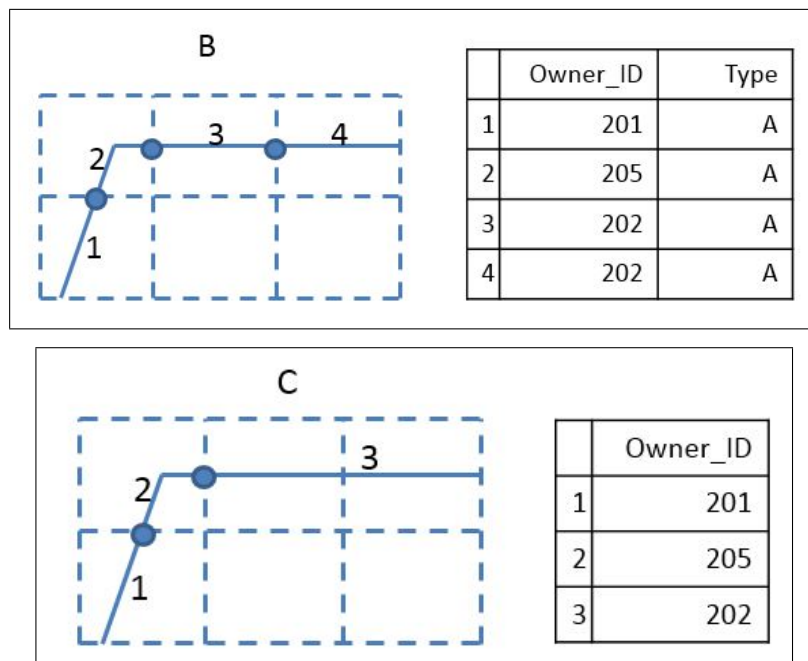


Name:

1. Consider the following diagram that represents operations over the input feature classes **Parcels** and **Facility** (the dotted line indicates the position of **Parcels**). Each feature is numbered, so it can be identified in the attribute table. Describe all features and attribute tables of output feature classes B and C.



One solution:

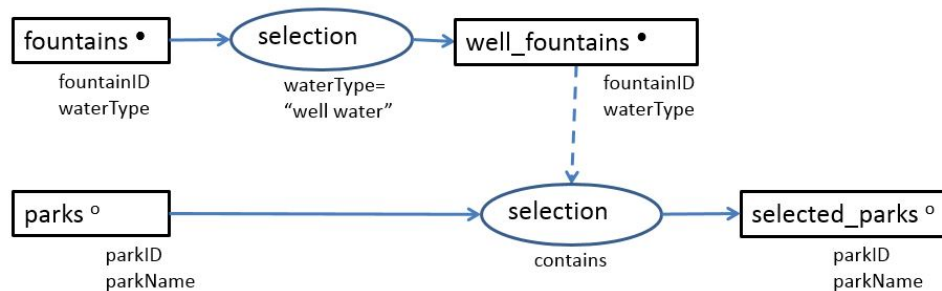


2. Suppose you have a GIS with the following feature classes:
- A polygon feature class with the location and names of public parks: `parks(parkID, parkName)`;
 - A point feature class with the location of water fountains, and the indication of the type of water which can be “well water” or “company water”: `fountains(fountainID, waterType)`;
 - A line feature class with the location and names of streets: `streets(streetID, streetName)`, where each street is represented by a unique geographic feature.

Write a diagram of GIS operations that provide a response to each of the following questions. The diagram should include the geometry of each feature class, the attributes of each table, the operations and their parameters if any; you can use operations like *selection*, *join*, *buffer*, or *intersection*.

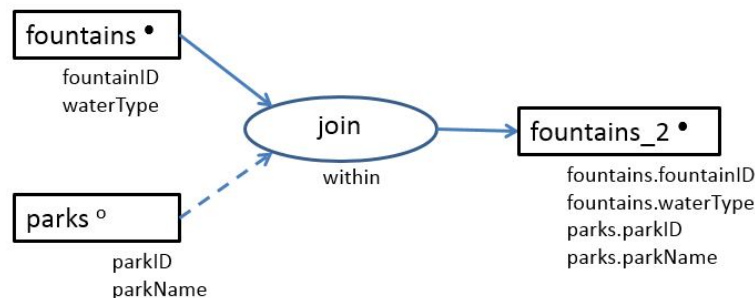
- (a) Determine the public parks that have fountains with well water;

One solution:



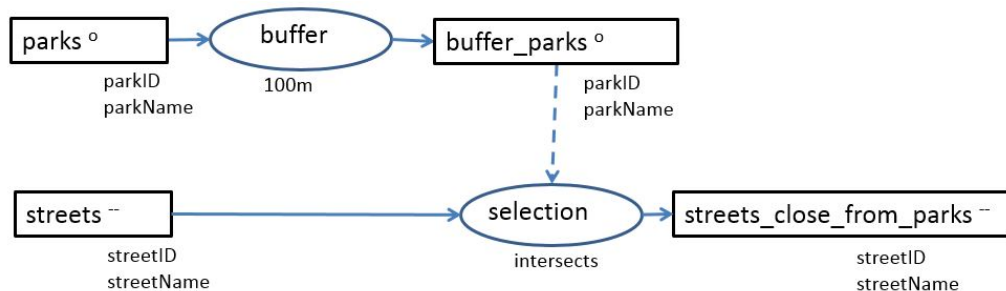
- (b) Associate to each fountain the name of the park it belongs to, i.e. produce a database table which has at least the following matching columns: **fountainID** and **parkName**;

One solution:



- (c) Determine the streets that pass 100m or less from public parks.

One solution:



3. To apply to EU subsidies, each farmer needs to fill an application. The application indicates the locations of the set of parcels of land that are explored by the farmer, and each crop the farmer intends to install in his land parcel (each parcel has one unique crop). Propose spatial and non spatial tables to structure the applicants information in a GIS. The GIS must store a 9-digit ID number for each applicant, the name of each applicant, the locations of all land parcels, a 2-letter crop code that indicates the crop to be installed on each parcel, and the name of each crop. The database should be normalized. Describe each table by a scheme of the form **TableName(attribute1,attribute2,...)** and indicate the domain of each attribute. For a table associated to a feature class, specify the geometry of the feature class.

One solution: The first database table **parcels** is associated to a polygon feature class where each polygon represents a parcel of land. The other two tables are non spatial database tables.

```

parcels(ID, applicantID, cropID)
applicants(applicantID, applicantName)
crops(cropID, cropName)
  
```

The domains of the attributes could be: ID[integer; 6], applicantID[integer; 9], cropID[string; 2], applicantName[string; 50], cropName[string; 20].