Datascript: Schema

```
{:name   {:db/unique :db.unique/identity}
 :alias  {:db/unique      :db.unique/value
          :db/cardinality :db.cardinality/many}
 :spouse {:db/cardinality :db.cardinality/one
          :db/valueType   :db.type/ref}
 :child  {:db/cardinality :db.cardinality/many
          :db/valueType   :db.type/ref}}
```

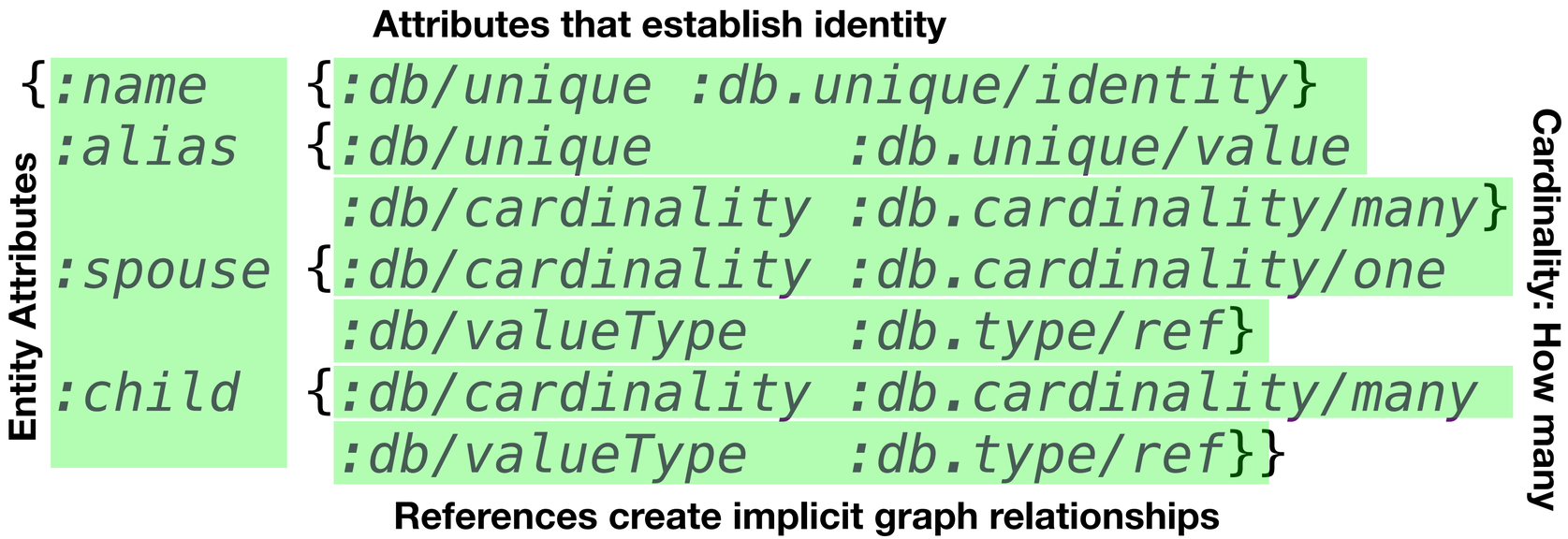
Entity Attributes

Attributes that establish identity

References create implicit graph relationships

Cardinality: How many

Datascript: Schema



Datascript: Data

```
[{:name "Peter Parker" :gender "M" :alias ["Spider-Man" "Spidey"]}
 {:name "Richard Parker" :gender "M" :spouse {:name "Mary Parker"}}
 {:name "Mary Parker" :gender "F" :spouse {:name "Richard Parker"}}
 {:name "Ben Parker" :gender "M" :spouse {:name "May Parker"}}
 {:name "May Parker" :gender "F" :spouse {:name "Ben Parker"}}
 {:name "Richard Parker" :child {:name "Peter Parker"}}
 {:name "Mary Parker" :child {:name "Peter Parker"}}
 {:child  [{:name "Ben Parker"}
           {:name "Richard Parker"}]
  :gender "M"}
 {:child [{:name "Ben Parker"}
          {:name "Richard Parker"}]
  :gender "F"}]
```

- Data is represented in its natural map form
- The underlying library will deconstruct the maps into datoms