

```
<map_stops_0> a rr:TriplesMap ;
```

```
rml:logicalSource [ a rml:LogicalSource ;  
  rml:referenceFormulation ql:CSV ;  
  rml:source "STOPS.csv"  
];
```

```
rr:subjectMap [ a rr:SubjectMap ;  
  rr:template "http://example.org/stops/{stop_id}"  
];
```

```
rr:predicateObjectMap [ a rr:PredicateObjectMap ;  
  rr:predicateMap [ a rr:PredicateMap ;  
    rr:constant wgs84:long  
  ] ;  
  rr:objectMap [ a rr:ObjectMap ;  
    rml:reference "stop_lon" ;  
    rr:datatype xsd:double ;  
    rr:termType rr:Literal  
  ] ;  
];
```

```
rr:predicateObjectMap [ a rr:PredicateObjectMap ;  
  rr:predicateMap [ a rr:PredicateMap ;  
    rr:constant <http://vocab.gtfs.org/terms#parentStation>  
  ] ;  
  rr:objectMap [ a rr:ObjectMap ;  
    rr:joinCondition [ rr:child "parent_station" ;  
      rr:parent "stop_id" ] ;  
    rr:parentTriplesMap <map_stops_0>  
  ] ;  
].
```

```
# <map_stops_0>  
CONSTRUCT { ?v4 wgs84:long ?v7 } {
```

```
SERVICE nose:rml.source { [ a rml:LogicalSource ;  
  rml:referenceFormulation ql:CSV ;  
  rml:source "STOPS.csv" ;  
  nose:output ?s0  
]}
```

```
BIND(IRI(concat("http://example.org/stops/",  
  encode_for_uri(str(nose:json.get(?s0, "stop_id")))) AS ?v4)
```

```
BIND(strdt(str(nose:json.get(?s0, "stop_lon")), xsd:double) AS ?v7)  
}
```

```
# <map_stops_0> -> <map_stops_0>  
CONSTRUCT { ?v4 <http://vocab.gtfs.org/terms#parentStation> ?s1_v4 } {
```

```
{ SELECT ?jc0 ?v4 {
```

```
SERVICE nose:rml.source { [ a rml:LogicalSource ;  
  rml:referenceFormulation ql:CSV ;  
  rml:source "STOPS.csv" ;  
  nose:output ?s0  
]}
```

```
BIND(nose:json.get(?s0, "parent_station") AS ?jc0)  
FILTER bound(?jc0)
```

```
BIND(IRI(concat("http://example.org/stops/",  
  encode_for_uri(str(nose:json.get(?s0, "stop_id")))) AS ?v4)
```

```
}}
```

```
{ SELECT ?jc0 ?s1_v4 {
```

```
SERVICE nose:rml.source { [ a rml:LogicalSource ;  
  rml:referenceFormulation ql:CSV ;  
  rml:source "STOPS.csv" ;  
  nose:output ?s1  
]}
```

```
BIND(nose:json.get(?s1, "stop_id") AS ?jc0)  
FILTER bound(?jc0)
```

```
BIND(IRI(concat("http://example.org/stops/",  
  encode_for_uri(str(nose:json.get(?s1, "stop_id")))) AS ?s1_v4)
```

```
}}
```

```
}
```