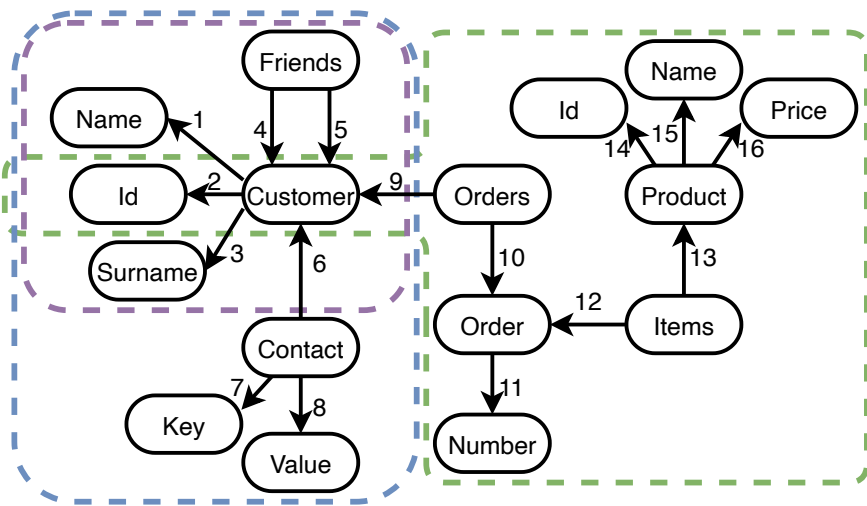


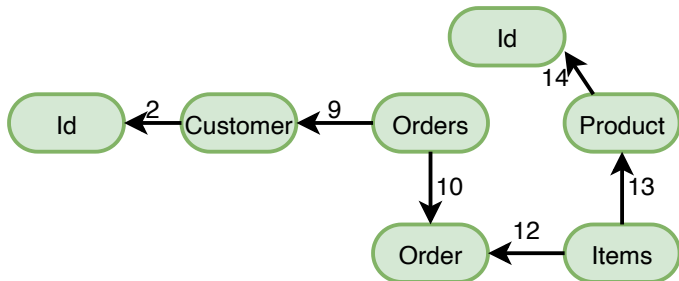
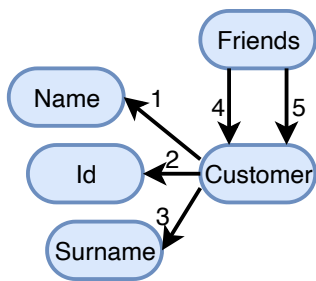
Phase I: Multi-model querying



```
SELECT {
  _:friendsOrders friend ?friendA ;
  friend ?friendB ;
  numProds COUNT(DISTINCT ?comProd)
    AS ?numProds .
  ?friendA name ?friendAName ;
  surname ?friendASurname .
  ?friendB name ?friendBName ;
  surname ?friendBSurname .
} WHERE {
  ?friendA -4/5 ?friendB .
  FILTER(?friendA < ?friendB)

  ?friendA -9/19/-12/-13 ?comProd ;
  1 ?friendAName ;
  3 ?friendASurname .
  ?friendB -9/19/-12/-13 ?comProd ;
  1 ?friendBName ;
  3 ?friendBSurname .
} ORDER BY ?numProds DESC
```

Phase II: Query decomposition into query parts



Phase III: Query parts translation into DSL

```
MATCH (a:Customer)-[:Friends]->(b:Customer)
RETURN a.id, a.name, a.surname, b.id, b.name, b.surname;
```

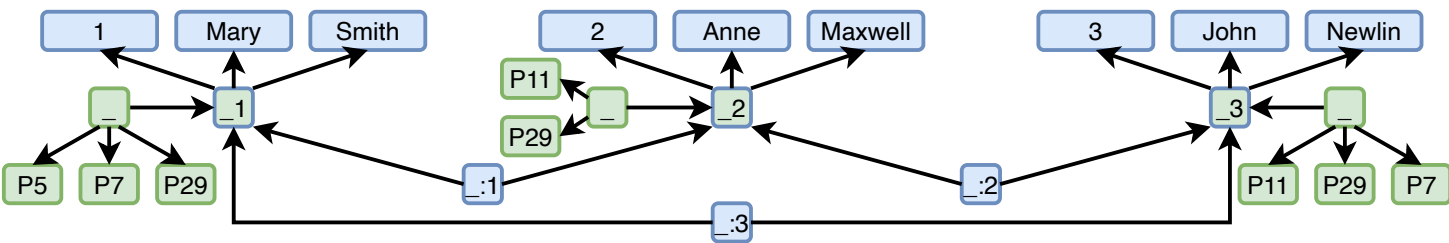
```
db.order.find(
  {},
  {_id: 0, customerId: 1, "items.productId": 1 } );
```

Phase IV: Evaluation of a translated query part (locally)

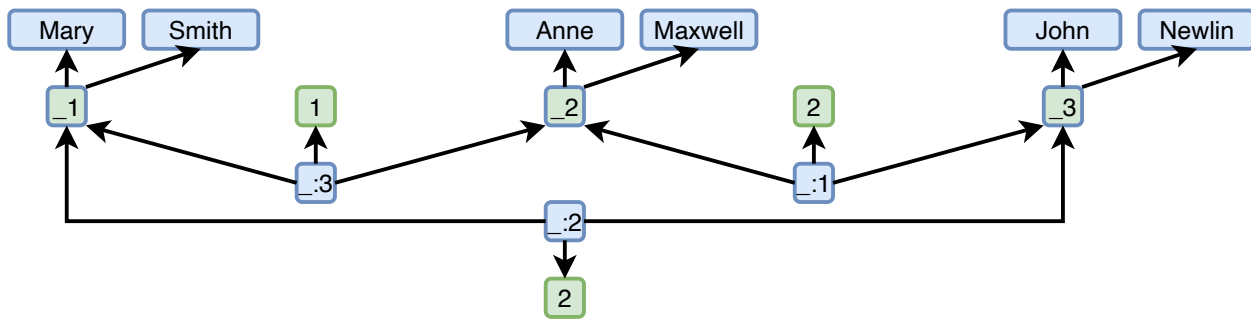
a.id	a.name	a.surname	b.id	b.name	b.surname
1	Mary	Smith	2	Anne	Maxwell
2	Anne	Maxwell	3	John	Newlin
3	John	Newlin	1	Mary	Smith

```
{ customerId: 1, items: [
  { productId: P5 }, { productId: P7 }, { productId: P29 } ] }
{ customerId: 2, items: [
  { productId: P11 }, { productId: P29 } ] }
{ customerId: 3, items: [
  { productId: P11 }, { productId: P29 }, { productId: P7 } ] }
```

Phase V: Transformation into unified representation



Phase VI: Finalization of query evaluation



Phase VII: Transformation into desired representation

friendAName	friendASurname	friendBName	friendBSurname	numProds
Anne	Maxwell	John	Newlin	2
John	Newlin	Mary	Smith	2
Mary	Smith	Anne	Maxwell	1