Type system in GraphQL

cngroup.dk

CN Group

Lenka Vondráčková

ReactNative Princess

vondrackova@cngroup.dk

CN Group

Scalars

- GraphQLInt
- GraphQLFloat
- GraphQLBoolean
- GraphQLString
- GrapQLID

```
hello: {
  type: GraphQLString,
  resolve: obj => 'world',
},
```

Resolve

- method invoked by the GraphQL execution engine when data are actually queried

- accepts 4 parameters:

    - source

    - args

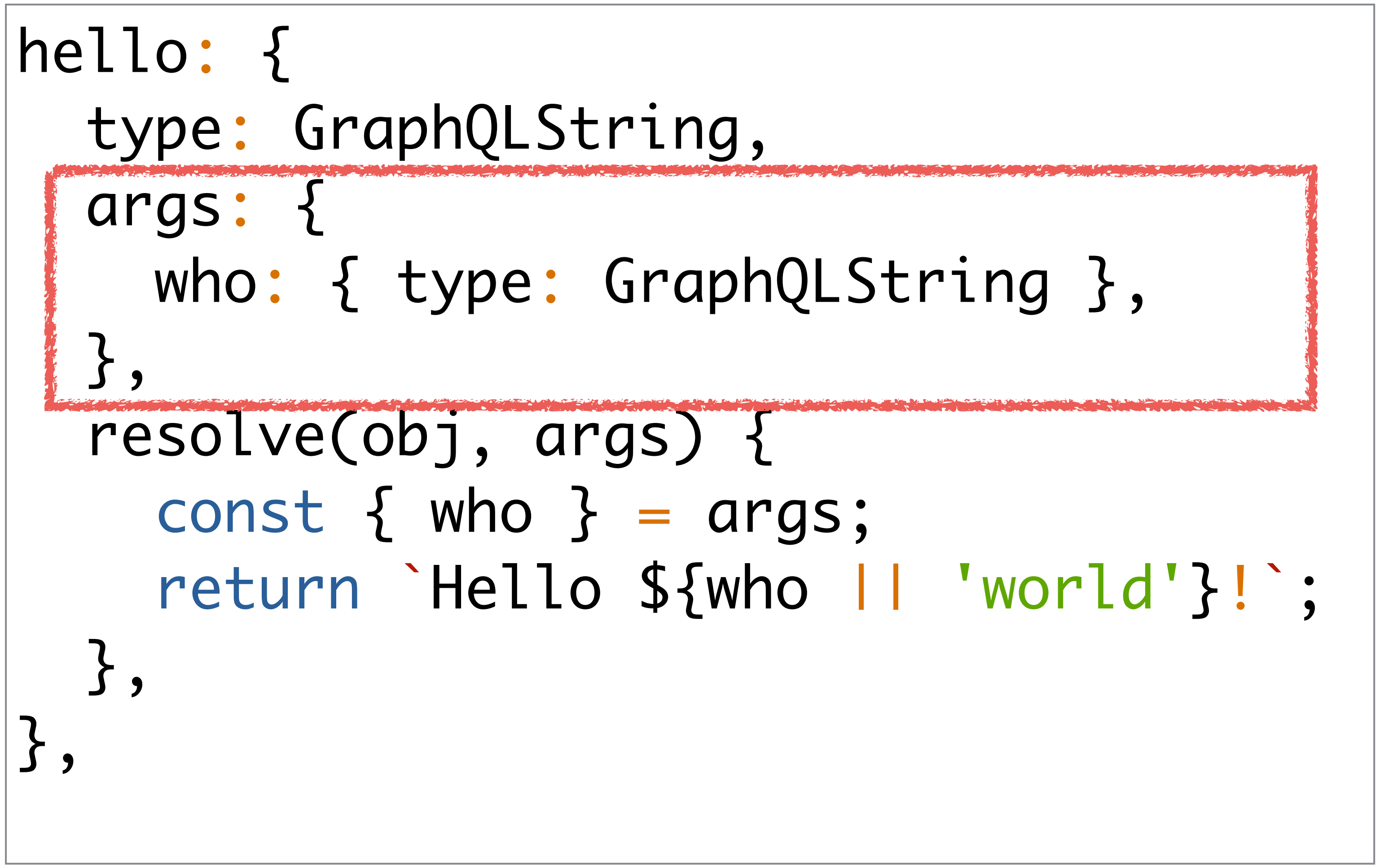
    - context

    - info

```
resolve(obj, args) {
  const { who } = args;
  return `Hello ${who || 'world'}!`;
},
```

Arguments

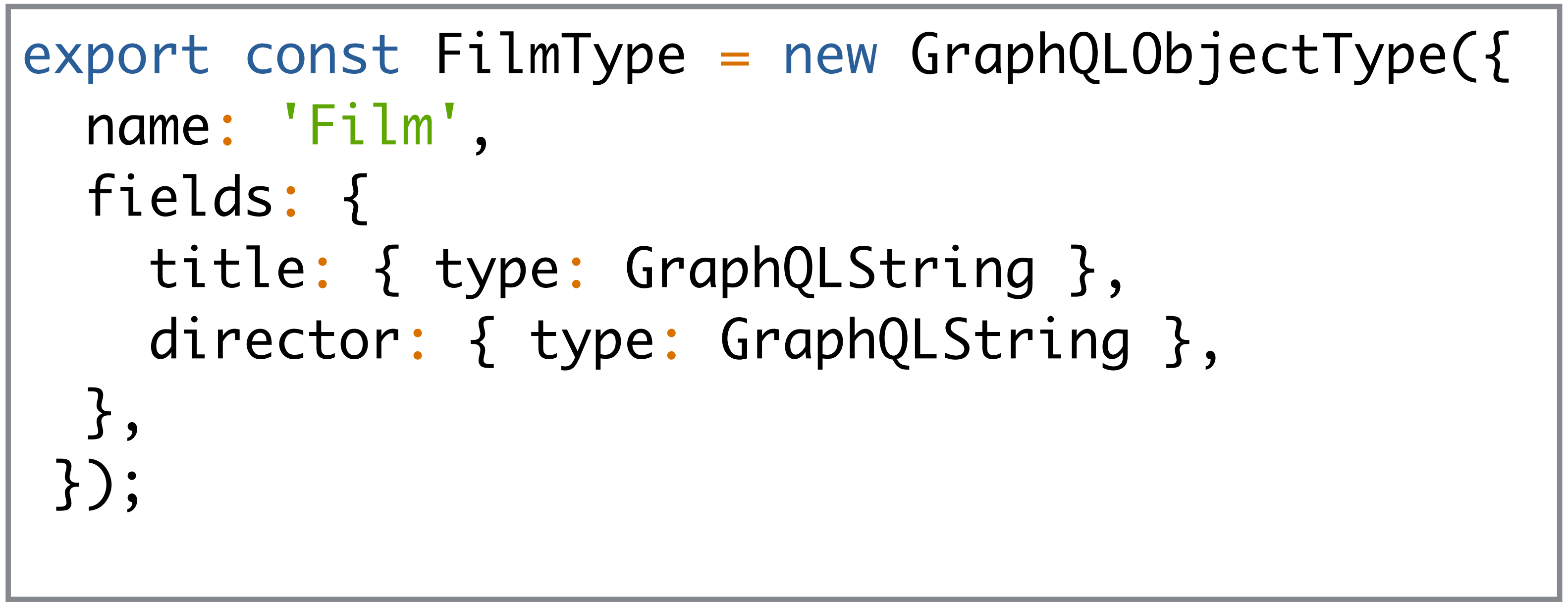
- second parameter accepted by resolve method
- enables to more specific queries
- has to be predefined



```
hello: {
  type: GraphQLString,
  args: {
    who: { type: GraphQLString },
  },
  resolve(obj, args) {
    const { who } = args;
    return `Hello ${who || 'world'}!`;
  },
},
```

Type Definitions

- Enables define your own composite types
  - GraphQLObjectType
  - GraphQLList

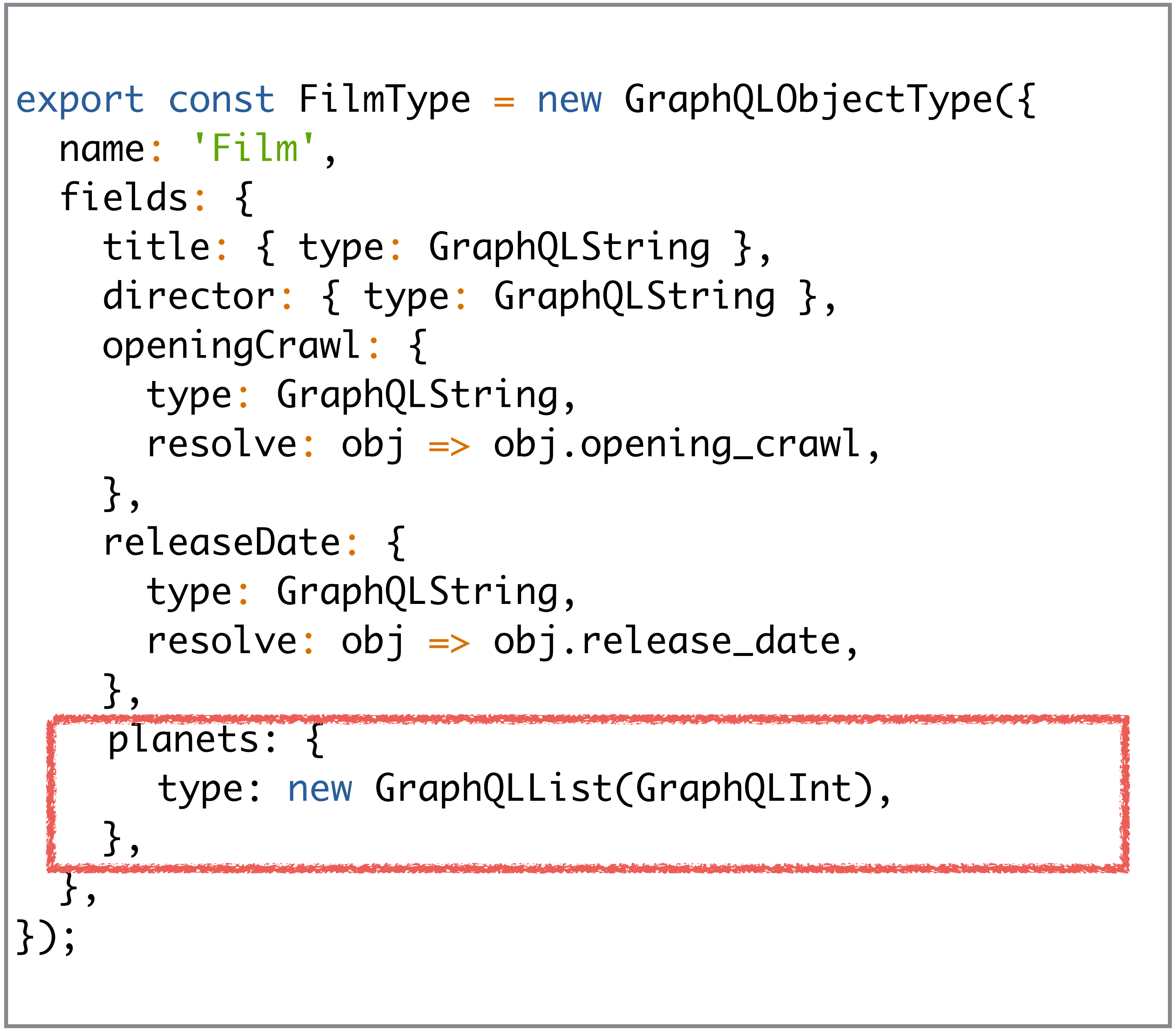


```
export const FilmType = new GraphQLObjectType({
  name: 'Film',
  fields: {
    title: { type: GraphQLString },
    director: { type: GraphQLString },
  },
 });
```

GraphQLObjectType ~ object

```
export const FilmType = new GraphQLObjectType({
  name: 'Film',
  fields: {
    title: { type: GraphQLString },
    director: { type: GraphQLString },
    openingCrawl: {
      type: GraphQLString,
      resolve: obj => obj.opening_crawl,
    },
    releaseDate: {
      type: GraphQLString,
      resolve: obj => obj.release_date,
    },
  },
});
```

GraphQLList ~ array

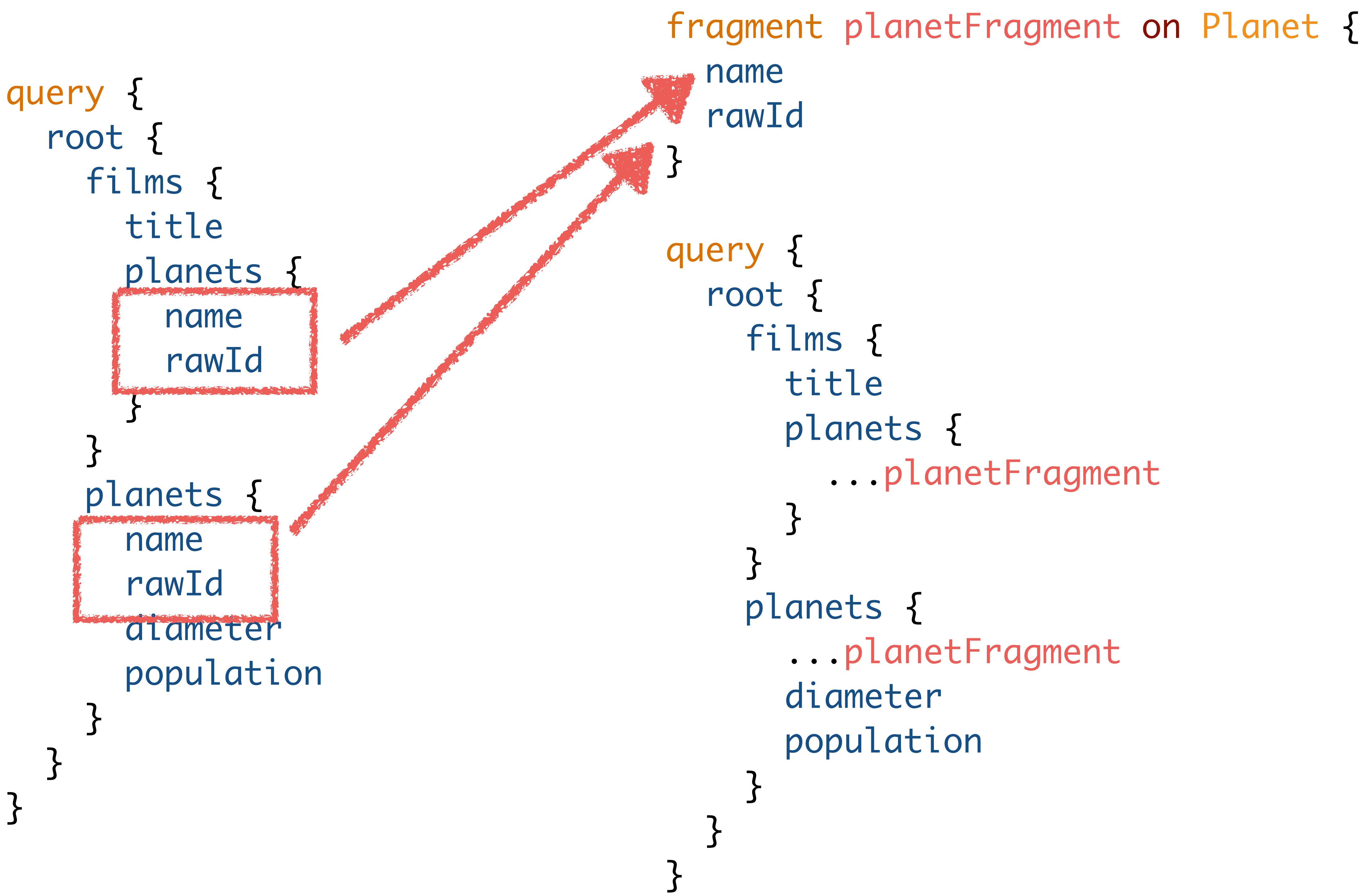


```
export const FilmType = new GraphQLObjectType({
  name: 'Film',
  fields: {
    title: { type: GraphQLString },
    director: { type: GraphQLString },
    openingCrawl: {
      type: GraphQLString,
      resolve: obj => obj.opening_crawl,
    },
    releaseDate: {
      type: GraphQLString,
      resolve: obj => obj.release_date,
    },
    planets: {
      type: new GraphQLList(GraphQLInt),
    },
  },
});
```

Fragments

http://graphql.org/docs/api-reference-type-system/

Fragments reduce repetition when querying same fields on different instances and are very useful with Relay



```
query {
  root {
    films {
      title
      planets {
        name
        rawId
      }
    }
    planets {
      name
      rawId
      diameter
      population
    }
  }
}
```

```
fragment planetFragment on Planet {
  name
  rawId
}

query {
  root {
    films {
      title
      planets {
        ...planetFragment
      }
    }
    planets {
      ...planetFragment
      diameter
      population
    }
  }
}
```