

Arcs Refinement Types for Dynamic Queries

Runtime refined generics

jopra@

Take the refinement system

```
schema Person
  name: Text
  age: Number
  pid: Text
  friend: &Person
```

```
particle PersonFetcher
  people: reads [Person {name, age}[age > 20]]
```

Add a query value

```
schema Person
  name: Text
  age: Number
  pid: Text
  friend: &Person
```

```
particle PersonFetcher
  people: reads [Person {name, age}[age > ?]] // Currently only supporting a single query type and query argument
```

Type check and code gen

```
schema Person
  name: Text
  age: Number
  pid: Text
  friend: &Person
```

```
particle PersonFetcher
  people: reads [Person {name, age}[age > ?]]
```

```
// Generates Kotlin code such as
class PeopleHandleType (name: String, age: Int) {
    ...
}
class PeopleHandle<PeopleHandleType...> {
    // query arg type is found using the refinement
    suspend query(Int query_arg): Set<EntityType> {
        ...
    }
}
```

Type check and code gen

```
schema Person
  name: Text
  age: Number
  pid: Text
  friend: &Person
```

```
particle PersonFetcher
  people: reads [Person {name, age}[age > ?]]
```

```
// Generates Kotlin code such as
class PeopleHandleType (name: String, age: Int) {
    ...
}
class PeopleHandle<PeopleHandleType...> {
    // query arg type is found using the refinement
    suspend query(Int query_arg): Set<EntityType> {
        ...
    }
}
```

```
// Generates Kotlin code for runtime filtering such as
fun pred(value: PeopleHandleType, arg: Int): Boolean
    = (value.age > arg)
```

```
// Generates SQLite query for database filtering such as
"SELECT ... FROM ... WHERE (entity_age.value > ${arg})"
```

Refinement system and interpreter:

<https://github.com/PolymerLabs/arcs/blob/master/src/runtime/refiner.ts>