Orysya Stus
A10743411
10.17.2017

DSE 250 HW 1

I. Here is a possible relational schema capturing the given ODL schema.

Note: underlined words are primary keys

Person (string ssn,
String name,
Date dob)

Boat (string name,
Int tonnage)

Ownership (int id,
String ssn,
String boat_name,
Date begin,
Date end)
Foreign key Ownership.ssn references Person.ssn
Foreign key Ownership.boat_name references Boat.name

RacesWon (string boat_name,
String race)
Foreign key RacesWon.boat_name references Boat.name

II. Express the following queries in OQL:

1. For the boats who won the "Americas Cup" title, return the (boat, owner) object pairs. The query result should have type **set<struct**{Boat boat, Person owner}>.

```
SELECT struct{boat: b, person: p}
FROM b in Boat, p in b.belongedTo.coOwners
WHERE b.racesWon = 'Americas Cup'
```

2. Find the boat(s) ever owned by "Jack Sparrow". The query result should have type set<Boat>.

```
SELECT b
FROM o in Ownership, b in o.Boat
WHERE o.coOwners.name = 'Jack Sparrow'
```

3. Now assume that the definition of class Person is enriched w/the declaration
relationship set<Ownership> ownerships inverse Ownership::coOwners;
And redo query II.2 exploiting this relationship

```
SELECTS b
FROM p in Person, b in p.ownerships.boat
WHERE p.name = 'Jack Sparrow'
```

Orysya Stus
A10743411
10.17.2017

4. Find the boat(s) most recently owned by “Jack Sparrow”. The query result should have type set<Boat>.

```
SELECT b
FROM p in Person, o in p.ownership
WHERE o.end = (
                    SELECT CASE o1.end WHERE null THEN o1.end ELSE MAX(o1.end)
                    FROM p1 in Person, o1 in p1.Ownership)
```

5. Dropping the assumption of point 3., find the owners (return the objects themselves) of all “Americas Cup” - winning boats.

```
SELECT p
FROM p in Person
WHERE for all w in (
                    SELECT b
                    FROM b in Boats
                    WHERE b.racesWon = ‘Americas Cup’): p in
w.belongedTo.coOwners
```

III. Express the queries II.1, II.2, II.4 and II.5 in QBE, on the schema of point I. Instead of returning objects, return the key of the corresponding entities.

II.1 QBE

racesWon	boat_name	race
	_b	Americas Cup

ownership	ssn	boat_name	begin	end
	P.	P._b		

II.2 QBE

person	ssn	name	dob
	_s	Jack Sparrow	

ownership	ssn	boat_name	begin	end
	_s	P.		

II.4 QBE

Phase I – Find end for all boats owned by Jack Sparrow, consider if end dates are Nulls

person	ssn	name	dob
	_s	Jack Sparrow	

ownership	ssn	boat_name	begin	end
	_s	_bn		Null

Orysya Stus
A10743411
10.17.2017

NULLS	boat_name	ssn
I.	_bn	_s

Phase II - Find end for all boats owned by Jack Sparrow, get not maximum end dates

Ownership	begin	end	boat_name	ssn
		e1	_bn1	_ssn
		e2	_bn2	_ssn

Condition
_e1 > _e2
_e1 not null
_e2 not null

NOTMAX	boatname	ssn
I.	_bn2	_ssn

Phase III – Get a list of boats that do not fall in Phase I or Phase II, the maximum

person	ssn	name	dob
	_s	Jack Sparrow	

ownership	ssn	boat_name	begin	end
	_s	_bn		

NOTMAX	boatname	ssn
┐	_bn	_s

NULLS	boat_name	ssn
┐	_bn	_s

MAX	boat_name	ssn
I.	_bn	_s

Phase IV – Get list of boat names ever owned by Jack Sparrow

MAX	boat_name	ssn
	_bn	_s

NULLS	boat_name	ssn
┐		_s

NULLS	boat_name	ssn
	_bn1	

RESULT	boat_name
I.	P._bn
	P._bn1

Orysya Stus
A10743411
10.17.2017

II.5 QBE

Phase I – Find boats who did not win

racesWon | boat_name | race
| _bn | Americas Cup

Ownership | ssn | boat_name | begin | end
ך | _s | _bn | |

Losers | ssn
I. | _s

Phase II – Get owners of winning boats

Losers | ssn
I. | _s

Person | ssn | name | dob
| P._s | |