For implementation an online retailer was chosen key-value database «Redis». The project was done on iOS platform in terminal without support of any other languages.

Installation process

1. Download the latest copy of Redis from official site (www.redis.io). In the project, redis was downloaded to Downloads (folder on macOS);
2. Execute commands on terminal:
    - `$ cd ~/Downloads/redis-3.2.9`
    - `$ make test` (to make sure Redis can complete a clean install on a computer)
    - `$ make`
3. Start up redis server and redis client using:
    - `MacBook-Pro-Nadezhda:~ Nadezhda$ redis-server`
    - `MacBook-Pro-Nadezhda:~ Nadezhda$ redis-cli`

Design a relational data model

As an example, model was built for online retailer. The basic concepts in an online store are customers (users) and product (items). Every customer has a basic set of data which will describe them: unique id, name, gender, age, etc. Similarly, a product has an id and a name.

Transform the data model to Redis data structures

Relational data model for Redis has a *key-value* structure (figure 1). KEY «user» include key-values as «UID» - unique id, «name» , «gender», «age». Unique KEY for users was chosen an email. KEY «item» consist of «UID» - unique id for an item, «category», «model», «price». Unique KEY for «item» is «item_UID». Technically Redis represent structure *key-key-value*. KEY «cart» is sorted set with relation {user:UID item:item_UID}. After all execute «payment».

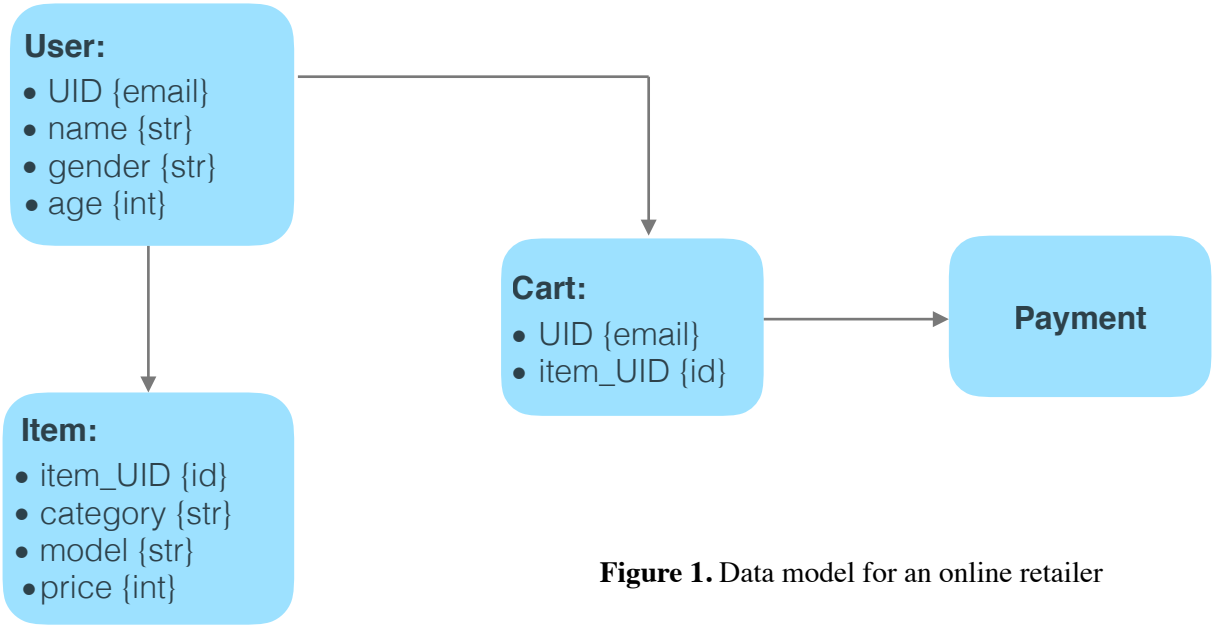

Figure 1. Data model for an online retailer

Creation of database

For further manipulation with data, need to fill a sufficient amount of sensible data into the Redis data structures. For database need to create:

- KEY «users»
- KEY «item»
- KEY «cart»
- payment

As was mention before firstly need to create a *key-value* for users. For registration new user in database use SADD and HSET commands.

Terminal:

```
127.0.0.1:6379> SADD users:names perkins@gmail.com
(integer) 1
127.0.0.1:6379> HSET users:perkins@gmail.com name "Quinn Perkins"
(integer) 1
127.0.0.1:6379> HSET users:perkins@gmail.com gender "female"
(integer) 1
127.0.0.1:6379> HSET users:perkins@gmail.com age 28
(integer) 1
```

As previously stated, unique id for users is their e-mail. Developer could choose unique KEY whatever he/she wants - telephone number, random id, address.

Afterwards was added 14 more users to database for further manipulations. To include necessary information to users using [HSET] or [HMSET] commands. [HMSET] using for multi fields, what can make filling faster.

Terminal:

```
127.0.0.1:6379> SADD users:names opope@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:opope@gmail.com name "Olivia Pope" gender "female" age «35"
OK
127.0.0.1:6379> SADD users:names dwyer@yahoo.com
(integer) 1
127.0.0.1:6379> HMSET users:dwyer@yahoo.com name "Lindsay Dwyer" gender "female" age 26
OK
127.0.0.1:6379> SADD users:names grantfitz@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:grantfitz@gmail.com name "Fitzgerald Grant" gender "male"
age 43
OK
127.0.0.1:6379> SADD users:names beebee@gmail.com
(integer) 1
HMSET users:beebee@gmail.com name "Cyrus Beene" gender "male" age 60
OK
127.0.0.1:6379> SADD users:names whelan@yahoo.com
(integer) 1
127.0.0.1:6379> HMSET users:whelan@yahoo.com name "Abby Whelan" gender "female" age 30
OK
```

```
127.0.0.1:6379> SADD users:names jakebake@yahoo.com
(integer) 1
127.0.0.1:6379> HMSET users:jakebake@yahoo.com name "Jake Ballard" gender "male" age 39
OK
127.0.0.1:6379> SADD users:names lizzinorth@mail.com
(integer) 1
127.0.0.1:6379> HMSET users:lizzinorth@mail.com name "Elizabeth North" gender "female"
age 41
OK
127.0.0.1:6379> SADD users:names huch@mail.com
(integer) 1
127.0.0.1:6379> HMSET users:huch@mail.com name "Diego Munoz" gender "male" age 40
OK
127.0.0.1:6379> SADD users:names marcwalk@mail.com
(integer) 1
127.0.0.1:6379> HMSET users:marcwalk@mail.com name "Marcus Walker" gender "male" age 36
OK
127.0.0.1:6379> SADD users:names darkman@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:darkman@gmail.com name "Harrison Wright" gender "male" age
35
OK
127.0.0.1:6379> SADD users:names rosenpower@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:rosenpower@gmail.com name "David Rosen" gender "male" age
42
OK
127.0.0.1:6379> SADD users:names primemellie@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:primemellie@gmail.com name "Mellie Grant" gender "female"
age 36
OK
127.0.0.1:6379> SADD users:names finch@gmail.com
(integer) 1
127.0.0.1:6379> HMSET users:finch@gmail.com name "Stephen Finch" gender "male" age 40
OK
127.0.0.1:6379> SADD users:names mrbergen@mail.com
(integer) 1
127.0.0.1:6379> HMSET users:mrbergen@mail.com name "Leo Bergen" gender "male" age 45
OK
```

To listing all all users in database:

```
127.0.0.1:6379> SMEMBERS users:names

 1) "beebee@gmail.com"
 2) "finch@gmail.com"
 3) "huch@mail.com"
 4) "whelan@yahoo.com"
 5) "jakebake@yahoo.com"
 6) "marcwalk@mail.com"
 7) "dwyer@yahoo.com"
 8) "darkman@gmail.com"
 9) "rosenpower@gmail.com"
10) "grantfitz@gmail.com"
11) "lizzinorth@mail.com"
12) "mrbergen@mail.com"
13) "opope@gmail.com"
14) "primemellie@gmail.com"
15) "perkins@gmail.com"
```

At the end database has 15 unique users. Every user has further information as «email», «name», «gender», «age». With command [HSET] possible to add any new field - date of birth, status, telephone number, address etc.

Next step to create an item database. «Items» consist of unique id, category, model, price. For working was chosen 10 items (table 1).

#	UID	Category	Brand	Model	Price	Currency
1	114d	mobile	Apple	iPhone 7	609	EUR
2	173d	mobile	Samsung	Galaxy S8	670	EUR
3	179c	mobile	Sony	Xperia Xcom	325	EUR
4	215g	laptop	Apple	MacBook Pro 15”	2349	EUR
5	210e	laptop	Asus	Zenbook UX51	1179	EUR
6	229e	laptop	Dell	XPS 13	1459	EUR
7	356a	accessories	mumbi	Protective case	7	EUR
8	353a	accessories	moozy	Flip case	12	EUR
9	385a	accessories	tuoya	Magnetic USB charger	8	EUR
10	302a	accessories	YGJ	Cover for Apple MacBook	20	EUR

Table 1. Lists of items for database

Unique id (UID) consist of numbers and letters. First number in sequence is a category (mobile - 1, laptop - 2, accessories - 3), second number is a brand (A:B - 1, C:D - 2, E:G - 3, H:J - 4, K:M - 5, N:P - 6, Q:S - 7, T:V - 8, W:X - 9, Y:Z - 0). Similarly schema was used in «model». Forth number is a price range, consist of letters what gives higher limit (0:50€ - a, 51:100€ - b, 101:500€ - c, 501:1000€ - d, 1001:1500€ - e, 1501:2000€ - f, 2001:2500€ - g).

Terminal:

```
127.0.0.1:6379> SADD item:id 114d
(integer) 1
127.0.0.1:6379> HMSET item:114d category "mobile" brand "Apple" model "iPhone 7" price
609 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 173d
(integer) 1
127.0.0.1:6379> HMSET item:173d category "mobile" brand "Samsung" model "Galaxy S8"
price 670 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 179c
(integer) 1
127.0.0.1:6379> HMSET item:179c category "mobile" brand "Sony" model "Xperia Xcom"
price 3325 currency "EUR"
```

```
OK
127.0.0.1:6379> SADD item:id 215g
(integer) 1
127.0.0.1:6379> HMSET item:215g category "laptop" brand "Apple" model "MacBook Pro
15''" price 2349 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 210e
(integer) 1
127.0.0.1:6379> HMSET item:210e category "laptop" brand "Asus" model "Zenbook UX51"
price 1179 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 229e
(integer) 1
127.0.0.1:6379> HMSET item:210e category "laptop" brand "Dell" model "XPS 13" price
1459 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 356a
(integer) 1
127.0.0.1:6379> HMSET item:356a category "accessories" brand "mumbi" model "Protective
case" price 7 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 353a
(integer) 1
127.0.0.1:6379> HMSET item:353a category "accessories" brand "moozy" model "Flip case"
price 12 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 385a
(integer) 1
127.0.0.1:6379> HMSET item:385a category "accessories" brand "tuoya" model "Magnetic
USB charger" price 8 currency "EUR"
OK
127.0.0.1:6379> SADD item:id 302a
(integer) 1
127.0.0.1:6379> HMSET item:302a category "accessories" brand "YGJ" model "Cover for
Apple MAcBook" price 20 currency "EUR"
OK
```

In database item's UID is id. For listing all product in database need to repeat command [SMEMBERS]:

Terminal:

```
127.0.0.1:6379> SMEMBERS item:id
 1) "173d"
 2) "229e"
 3) "179c"
 4) "353a"
 5) "356a"
 6) "210e"
 7) "302a"
 8) "215g"
 9) "114d"
10) "385a"
```