

# Домашнее задание 2

---

Хамид Карим

---

Не совсем понял, что конкретно сдавать, поэтому есть и код, и картинка.

---

## Задание 1

```
@startuml

entity Book {
    * ISBN: text
    ---
    year: number
    name: text
    author: text
    publisher_id: number <<FK>>
    pages_amount: number
}

entity Publisher {
    * uuid: UUID <<generated>>
    ---
    name: text
    address: text
}

entity Copy {
    * uuid: UUID <<generated>>
    ---
    * copy_number: number
    * book_isbn: text
    shelf_position: text
}

entity Category {
    * uuid: UUID <<generated>>
    ---
    parent_category_uuid: UUID <<FK>>
    name: text
}

entity Reader {
    * uuid: UUID <<generated>>
    ---
    name: text
    surname: text
    address: text
}
```

```

    burthdate: date
  }

entity Borrow {
  * uuid: UUID <<generated>>
  ---
  * copy_uuid: UUID <<FK>>
  * reader_uuid: UUID <<FK>>
  return_date: date
}

Publisher ||--o{ Book: publishes
Book ||--o{ Copy: has
Book }o--|{ Category: categorized
Category |o--o{ Category: has parent category
Reader ||--o{ Borrow: borrowed by
Borrow }o--|| Copy: borrows

@enduml

```

- 
- Да, в Copy можно было бы сделать

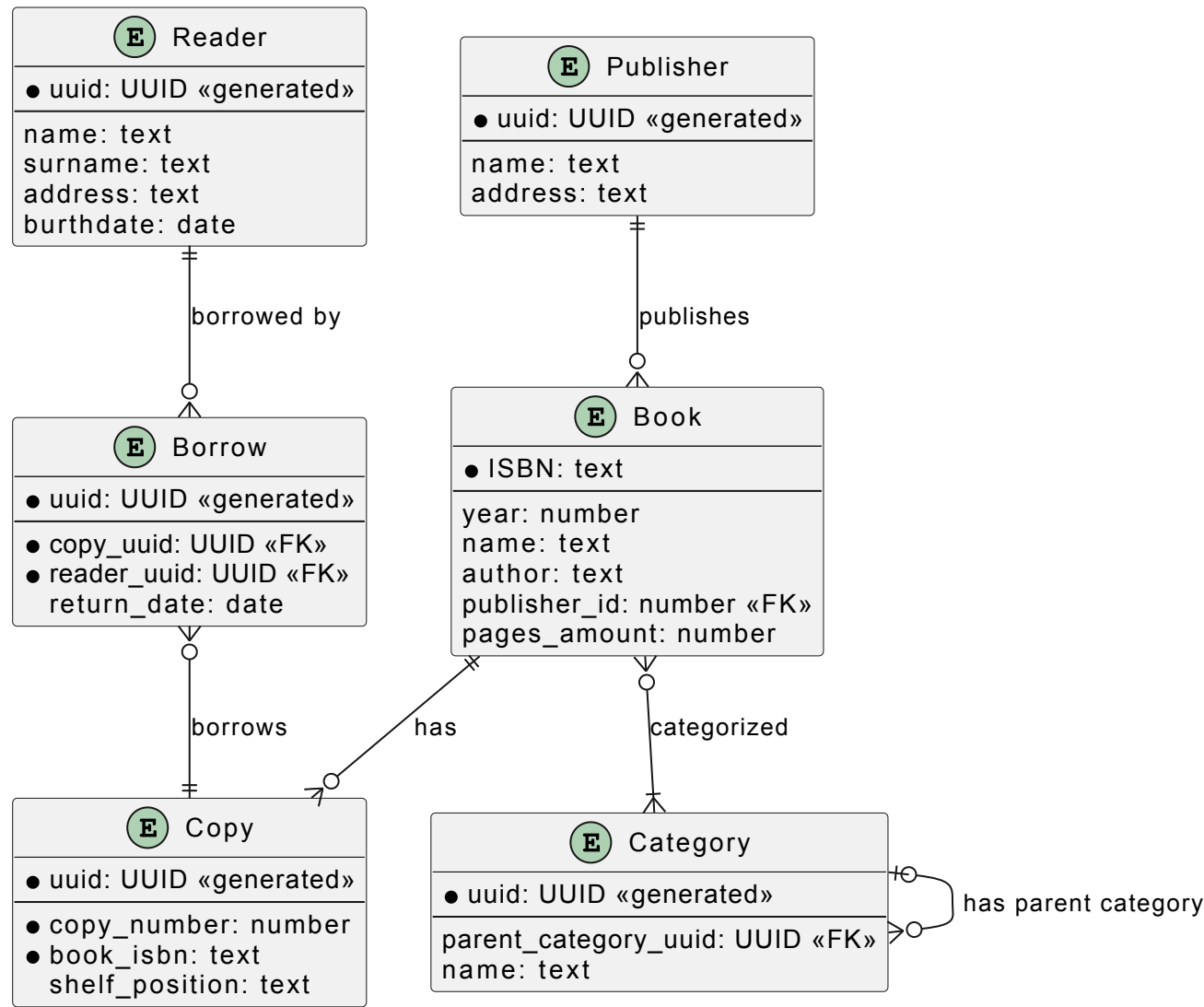
```

entity Copy {
  * copy_number: number
  * book_isbn: text
  ---
  shelf_position: text
}

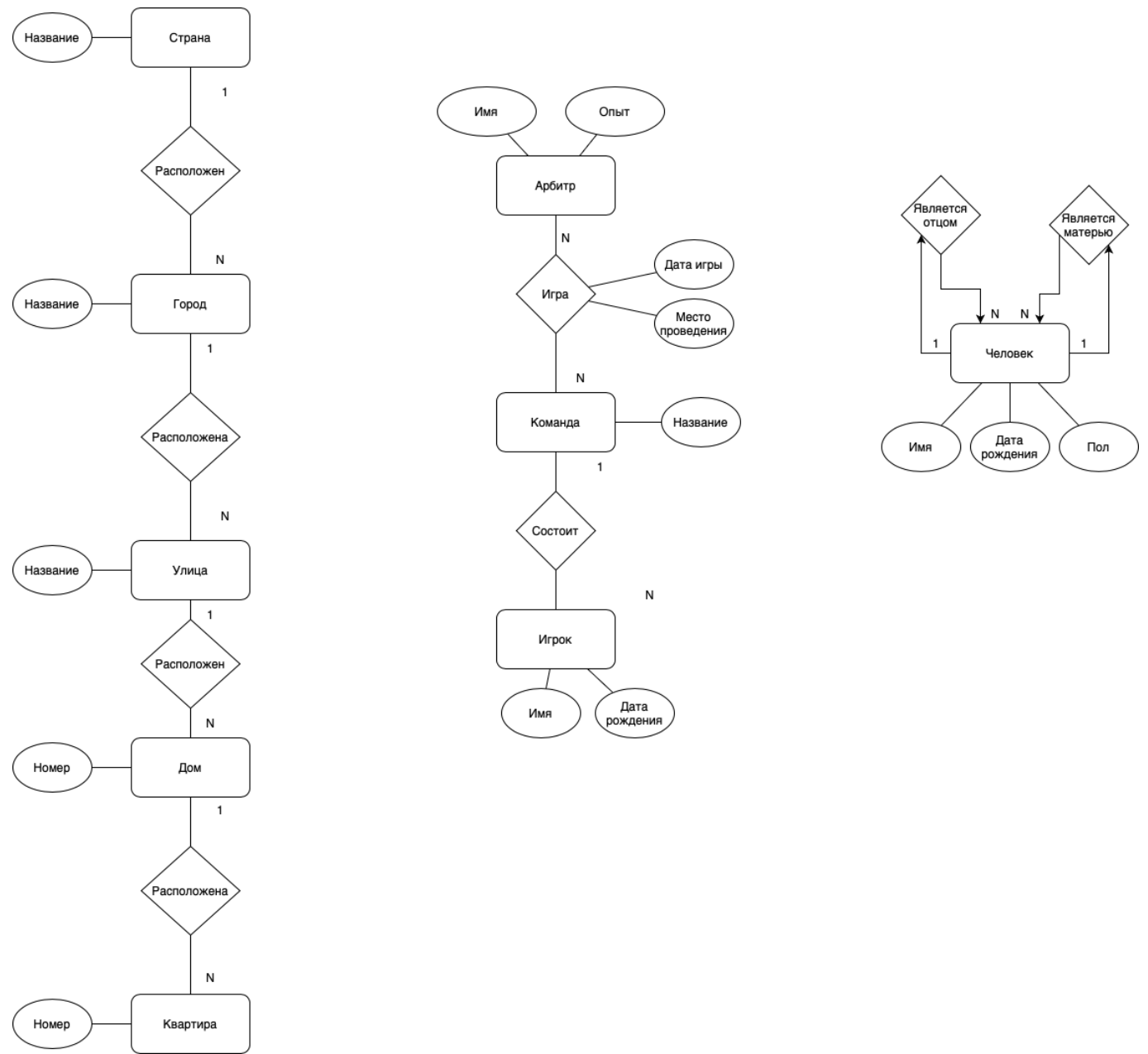
```

, но тогда и ссылаться из Borrow надо на два значения, поэтому я сделал uuid и обязательные поля copy\_number и book\_isbn

- По факту, Borrow является таблицей M2M связи с доп информацией



Задание 2



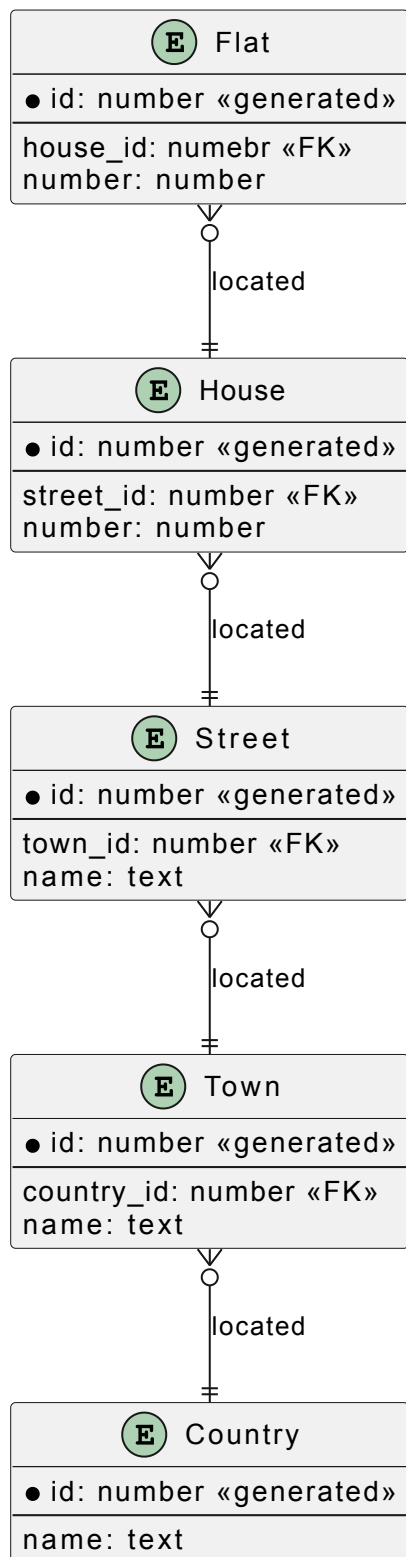
Задание 3

Пункт 1

```
@startuml
entity Flat {
  * id: number <<generated>>
  ---
  house_id: numebr <<FK>>
  number: number
}
entity House {
  * id: number <<generated>>
  ---
  street_id: number <<FK>>
  number: number
}
entity Street {
```

```
* id: number <<generated>>
---
town_id: number <<FK>>
name: text
}
entity Town {
  * id: number <<generated>>
  ---
  country_id: number <<FK>>
  name: text
}
entity Country {
  * id: number <<generated>>
  ---
  name: text
}

Flat }o--|| House: located
House }o--|| Street: located
Street }o--|| Town: located
Town }o--|| Country: located
@enduml
```



## Пункт 2

```

@startuml

entity Player {
    * id: number <<generated>>
    ---
    name: text
    birthdate: date
}
  
```

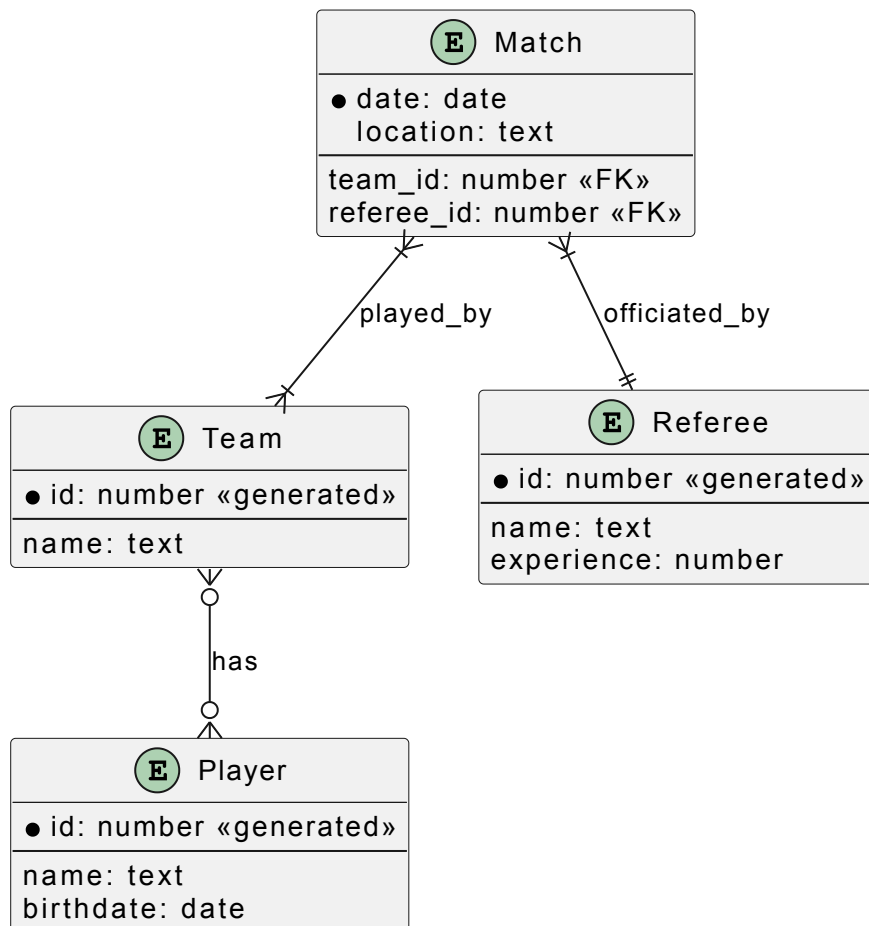
```
entity Team {
  * id: number <<generated>>
  ---
  name: text
}

entity Referee {
  * id: number <<generated>>
  ---
  name: text
  experience: number
}

entity Match {
  * date: date
  location: text
  ---
  team_id: number <<FK>>
  referee_id: number <<FK>>
}

Team }o--o{ Player: has
Match }|--|{ Team: played_by
Match }|--|| Referee: officiated_by

@enduml
```



### Пункт 3

```

@startuml

entity Person {
    * id: number <<generated>>
    ---
    name: text
    birthdate: date
    gender: text
    mother_id: number <<FK>>
    father_id: number <<FK>>
}

Person ||--o{ Person: isMotherOf
Person ||--o{ Person: isFatherOf

@enduml

```



