



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления» _____

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии» _____

ОТЧЁТ ПО ЛАБОРАТОРНОЙ **РАБОТЕ**

Студенты: Лозовска Карина и Сарварова Мария

Группы: ИУ9-51Б, ИУ9И-54Б

Преподаватель: А.Н. Непейвода

2022 г.

СОДЕРЖАНИЕ

	Техническое задание	3
1	Постановка задачи	4
2	Индивидуальное задание	5
	2.1 Входная грамматика	5
	2.2 Выходные данные	5
	2.3 Листинг	6
3	Пример работы программы на тестовых данных	12
4	Заключение	16

Техническое задание

Условие задания состоит из нескольких пунктов:

1. Составление грамматики для описания объекта
2. Составление документации: как задавать пользовательский синтаксис, а также как записывать требуемый объект (или параметры для генерации)
3. Считывание значений параметров из отдельного файла "syntax.txt"
4. Построение объекта по данным, учитывая синтаксис
5. Обобщение полученных результатов и навыков, составление отчёта по практике

1 Постановка задачи

- Необходимо предложить грамматику описания сущностей, атрибутов и связей между ними с учётом кардинальностей. Параметризованными токенами грамматиками могут выступать, например:
 1. обозначения кардинальностей и типов ключей
 2. способы группировки атрибутов, относящихся к объектам (например, синтаксис скобочной структуры, ограничивающей список атрибутов, относящихся к одному и тому же объекту)
- В заданном синтаксисе из входного файла читается описание ER-диаграммы
- Результатом должна быть ER-диаграмма и реляционная диаграмма, полученная из неё посредством процедуры преобразования. Имена порождённых при преобразовании новых сущностей должны генерироваться автоматически

Также в лабораторной работе автоматически извлекается таблица кардинальностей из описания модели.

2 Индивидуальное задание

2.1 Входная грамматика

Входные данные: текстовый файл, в котором прописаны сущности. У них указаны названия, primary key, foreign key, атрибуты и возможные отношения ребёнка, родителя или что-то иное. Входную грамматику можно увидеть ниже:

$$\begin{aligned} [Grammar] &\rightarrow [Entities] \\ [Entities] &\rightarrow [Entity] \mid [Entity]; [Entities] \\ [Entity] &\rightarrow [Name] :: [PK]; [FK] \mid NULL; [Attributes] \mid NULL; Parent : \\ [Parent] \mid NULL; Child : [Child] \mid NULL; OtherRelations : \\ [OtherRelations] \mid NULL \\ [Name] &\rightarrow [a - zA - Z] [Str] \\ [PK] &\rightarrow [a - zA - Z] [Str], [PK] \mid [a - zA - Z] [Str] \\ [FK] &\rightarrow [a - zA - Z] [Str], [FK] \mid [a - zA - Z] [Str] \\ [Attributes] &\rightarrow [Attribute], [Attributes] \mid [Attribute] \\ [Attribute] &\rightarrow [a - zA - Z] [Str] \\ [Parent] &\rightarrow [Name] ([Cardinality]), [Parent] \mid [Name] ([Cardinality]) \\ [Child] &\rightarrow [Name] ([Cardinality]), [Child] \mid [Name] ([Cardinality]) \\ [OtherRelations] &\rightarrow [Name] ([Cardinality]), [OtherRelations] \mid \\ [Name] &([Cardinality]) \\ [Cardinality] &\rightarrow 1 \mid 0 - M \mid 1 - M \\ [Str] &\rightarrow [a - zA - Z0 - 9_] [Str] \mid \epsilon \end{aligned}$$

2.2 Выходные данные

ER-диаграмма, реляционная диаграмма, полученная из неё посредством преобразования, таблица кардинальности из описания модели

Тестовые данные: Пример случайной модели

2.3 Листинг

```
1 class Entity:
2     def __init__(self, Name):
3         self.Name = Name
4         self.PK = []
5         self.FK = []
6         self.Attributes = []
7         self.Parent = {}
8         self.Child = {}
9         self.OtherRelations = {}
```

Листинг 1: Класс сущностей

Данный отрывок кода содержит функцию парсинга исходного файла.

Определение. Парсинг – индексирования информации с последующей конвертацией ее в иной формат или даже иной тип данных.

```
1 x = x.split('::') # Имя сущности
2 for i, y in enumerate(x):
3     x[i] = x[i].strip(' ').split(';')
4 Ent = Entity(x[0][0])
5
6 for y in (x[1][0].split(",")): # Primary Key
7     Ent.PK.append(y.strip(' '))
8
9 for y in (x[1][1].split(',')): # Foreign Key
10     if y.find('NULL') != -1: break
11     Ent.FK.append(y.strip(' '))
12
13 for y in (x[1][2].split(',')): # Attributes
14     Ent.Attributes.append(y.strip(' '))
15
16 for y in (x[1][3].replace('Parent : ', ' ').split(',')): # Parent
17     if y.find('NULL') != -1: break
18     y = y.strip(' ').replace('(', ' ').replace(')', ' ')
19     Ent.Parent[(y.split(' ')[0]) = (y.split(' ')[1])
20
21 for y in (x[1][4].replace('Child : ', ' ').split(',')): # Child
22     if y.find('NULL') != -1: break
23     y = y.strip(' ').replace('(', ' ').replace(')', ' ')
24     Ent.Child[(y.split(' ')[0]) = (y.split(' ')[1])
```

```

25
26 for y in (x[1][5].replace(' OtherRelations : ', ' ').split(', ')): # Other Relations
27     if y.find(' NULL ') != -1: break
28     y = y.strip(' ').replace('(', ' ').replace(')', ' ')
29     Ent.OtherRelations[(y.split(' ')[0])] = (y.split(' ')[1])
30
31 return Ent

```

В лабораторной работе была использована библиотека *JSON*, с помощью которой мы преобразовали наши входные данные в формат *JSON*. А далее уже из этого файла с помощью *rdot* получаем графики ег-модели и реляционной модели.

Далее заполняем информацию об атрибутивных сущностях и об отношениях между сущностями (Листинг 3). Но сначала преобразуем обозначения кардинальности к нужному виду (Листинг 2):

$0 - M \rightarrow *$

$1 - M \rightarrow +$

$1 \rightarrow 1$

```

1 for j, i in enumerate(Entities):
2     for k, v in i.OtherRelations.items():
3         if v == "1-M": Entities[j].OtherRelations[k] = "+"
4         elif v == "0-M": Entities[j].OtherRelations[k] = "*"
5
6 for j, i in enumerate(Entities):
7     for k, v in i.Parent.items():
8         if v == "1-M": Entities[j].Parent[k] = "+"
9         elif v == "0-M": Entities[j].Parent[k] = "*"
10
11 for j, i in enumerate(Entities):
12     for k, v in i.Child.items():
13         if v == "1-M": Entities[j].Child[k] = "+"
14         elif v == "0-M": Entities[j].Child[k] = "*"

```

Листинг 2: Преобразование обозначения кардинальности к нужному виду

```

1      for num, ent in enumerate(Entities):
2  for k, v in ent.Parent.items():
3      for i in Entities:
4          if i.Name == k:
5              for k1, v1 in i.Child.items():
6                  if k1 == ent.Name:
7                      result = v1
8                      dic['relations'].append(str(k) + ":" + i.PK[0] + " " + str(v) + "--" +
9                          str(result) + " " + ent.Name + ":" + ent.FK[0])
10                     break
11
12 for k, v in ent.OtherRelations.items():
13     for num1, i in enumerate(Entities):
14         if (i.Name == k) & (num1 > num):
15             for k1, v1 in i.OtherRelations.items():
16                 if k1 == ent.Name:
17                     result = v1
18                     dic['relations'].append(ent.Name + ":" + ent.PK[0] + " " + result + "--" +
19                         v + " " + k + ":" + i.PK[0])
20                     break

```

Листинг 3: Заполнение информации

```

1      dic = to_json_file(Entities, 1)
2  with open("diagram1.json", "w") as write_file:
3      json.dump(dic, write_file, indent=4)
4      command = "python3 --version"
5
6  os.system("erdot diagram1.json")
7  os.system("dot diagram1.dot -Tpng -o diagram1.png")

```

Листинг 4: Модель сущность-связь

Для преобразования ER-модели в реляционную мы выделяем обозначение *FK* и *PK*, а также добавляем новые сущности. Для этого проверяем сущности на связь многие ко многим, а затем добавляем связи с новой сущностью в изначальные.


```

1         for i, e in enumerate(Entities):
2     if e.Name == z1[0]:
3         if y[3] == '+': Entities[i].Child[ent.Name] = '1-M'
4         if y[3] == '*': Entities[i].Child[ent.Name] = '0-M'
5         e.OtherRelations.pop(z2[0])
6         continue
7     if e.Name == z2[0]:
8         if y[0] == '+': Entities[i].Child[ent.Name] = '1-M'
9         if y[0] == '*': Entities[i].Child[ent.Name] = '0-M'
10        e.OtherRelations.pop(z1[0])
11        continue

```

Листинг 5: Добавление связей в реляционной модели

Для дополнительного задания, построения таблицы кардинальности, была использована библиотека *tabulate*. В два цикла мы сначала проходим по отношениям ребёнка - родителя, а потом уже по остальным сущностям. И сразу заполняем таблицу нужными данными для её построения.

```

1      if i.Child[ent.Name] == '1':
2          if ent.Parent[i.Name] == '1':
3              newdata.append('1:1')
4              newdata.append('M-O')
5          if ent.Parent[i.Name] == '0-M':
6              newdata.append('1:N')
7              newdata.append('M-O')
8          if ent.Parent[i.Name] == '1-M':
9              newdata.append('N:1')
10             newdata.append('M-M')
11     if i.Child[ent.Name] == '0-M':
12         if ent.Parent[i.Name] == '1':
13             newdata.append('N:1')
14             newdata.append('O-M')
15         if ent.Parent[i.Name] == '0-M':
16             newdata.append('M:N')
17             newdata.append('O-O')
18         if ent.Parent[i.Name] == '1-M':
19             newdata.append('M:N')
20             newdata.append('O-M')
21     if i.Child[ent.Name] == '1-M':
22         if ent.Parent[i.Name] == '1':
23             newdata.append('1:N')
24             newdata.append('M-M')
25         if ent.Parent[i.Name] == '0-M':
26             newdata.append('M:N')
27             newdata.append('M-O')
28         if ent.Parent[i.Name] == '1-M':
29             newdata.append('M:N')
30             newdata.append('M-M')
31     data.append(newdata)

```

Листинг 6: Заполнение отношений родитель – ребёнок

```

1      if i.OtherRelations[ent.Name] == '1':
2          if ent.OtherRelations[i.Name] == '1':
3              newdata.append('1:1')
4              newdata.append('M-O')
5          if ent.OtherRelations[i.Name] == '0-M':
6              newdata.append('1:N')
7              newdata.append('M-O')
8          if ent.OtherRelations[i.Name] == '1-M':
9              newdata.append('1:N')
10             newdata.append('M-M')
11     if i.OtherRelations[ent.Name] == '0-M':
12         if ent.OtherRelations[i.Name] == '1':
13             newdata.append('1:N')
14             newdata.append('O-M')
15         if ent.OtherRelations[i.Name] == '0-M':
16             newdata.append('M:N')
17             newdata.append('O-O')
18         if ent.OtherRelations[i.Name] == '1-M':
19             newdata.append('M:N')
20             newdata.append('O-M')
21     if i.OtherRelations[ent.Name] == '1-M':
22         if ent.OtherRelations[i.Name] == '1':
23             newdata.append('1:N')
24             newdata.append('M-M')
25         if ent.OtherRelations[i.Name] == '0-M':
26             newdata.append('M:N')
27             newdata.append('M-O')
28         if ent.OtherRelations[i.Name] == '1-M':
29             newdata.append('M:N')
30             newdata.append('M-M')
31     data.append(newdata)

```

Листинг 7: Заполнение других отношений между сущностями

3 Пример работы программы на тестовых данных

Тестовые входные данные:

Customer :: Phone; NULL; FirstName, LastName, Email, City; Parent : NULL;
Child : NULL; OtherRelations : Order (0-M)
BookStore :: BookstoreName; NULL; Email, URL, Phone; Parent : NULL;
Child : NULL; OtherRelations : Order (0-M)
Order :: OrderNumber; NULL; PaymentType, MakingOrderDate, DeliveryDate;
Parent : NULL; Child : OrderLineBook (1-M); OtherRelations : Customer (1),
BookStore (1)
OrderLineBook :: LineNumber; OrderNumber, ISBN; Quantity, ExtendedPrice;
Parent : Order (1), Book (1); Child : NULL; OtherRelations : Book (1)
Book :: ISBN; NULL; Title, Genre, PublishingHouse, PublishingYear, Price;
Parent : NULL; Child : OrderLineBook (1-M); OtherRelations : Author (1-M)
Author :: AuthorID; NULL; FirstName, LastName, DeathYear, Country; Parent : NULL;
Child : NULL; OtherRelations : Book (0-M)

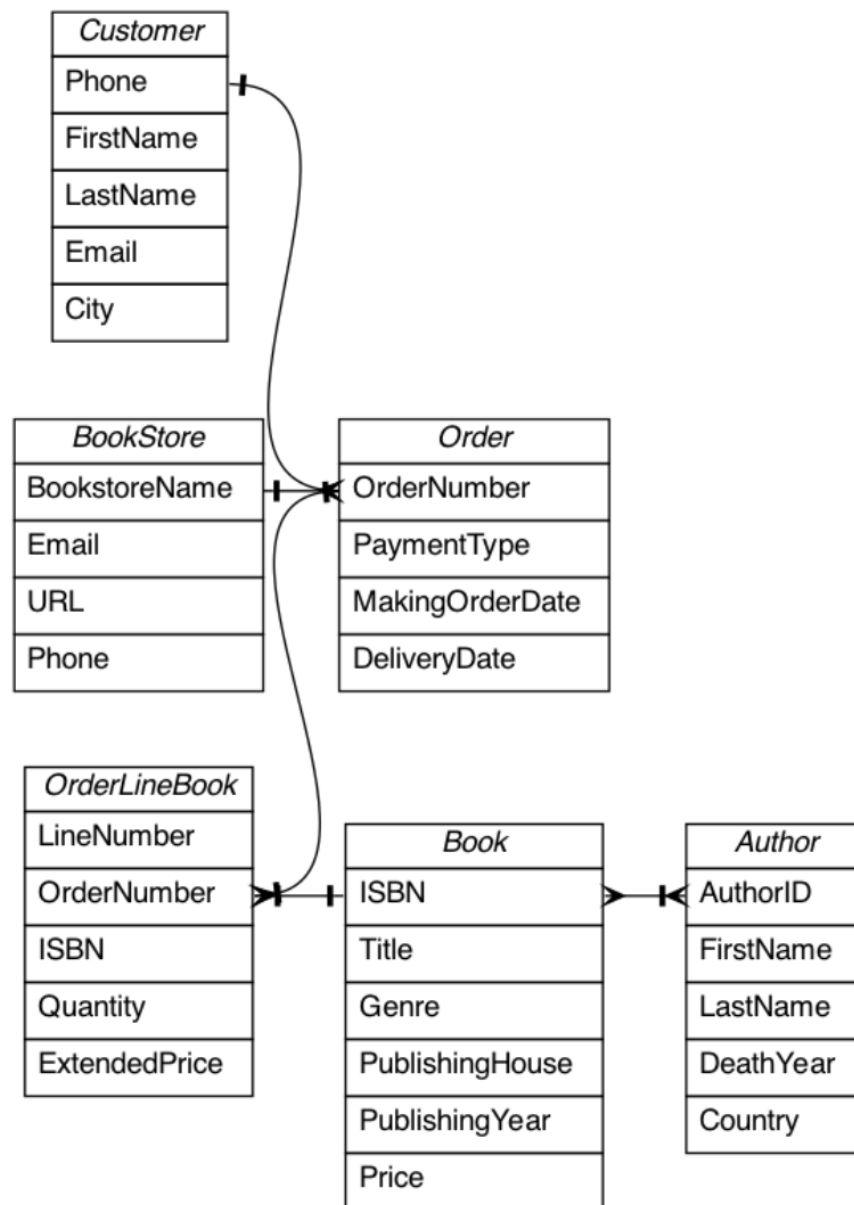


Рисунок 1 — ER-модель

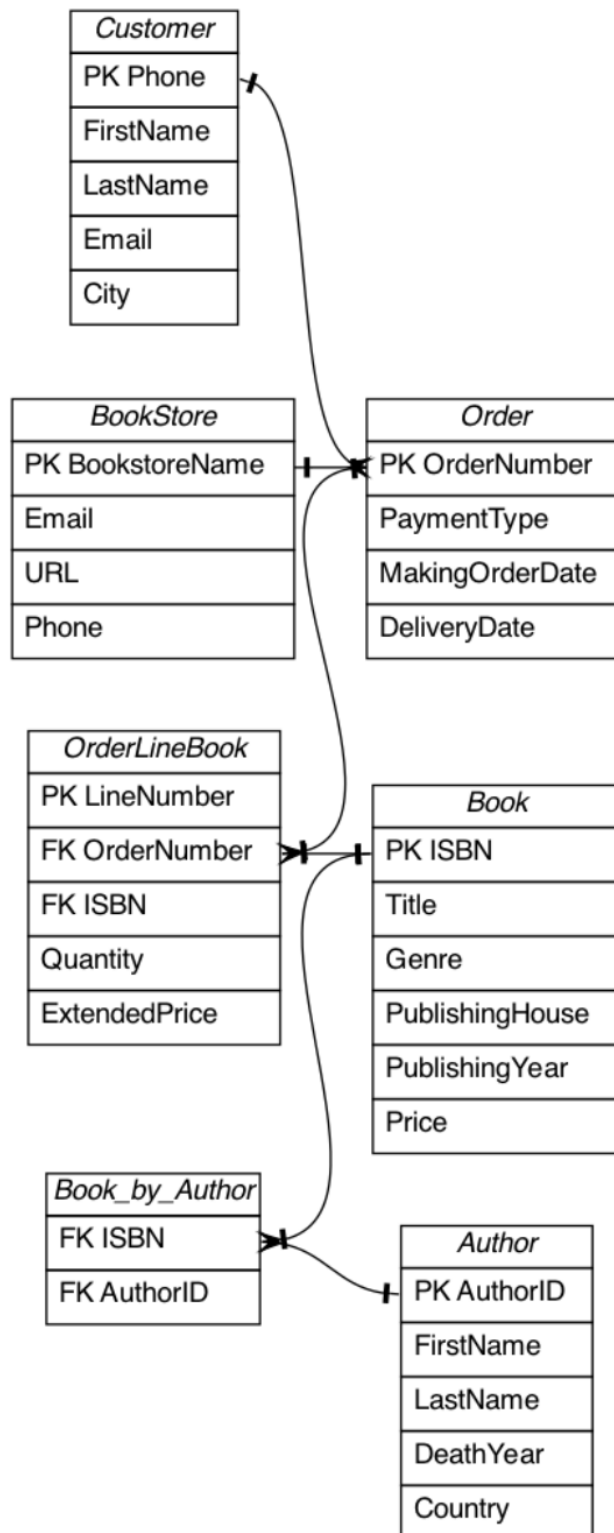


Рисунок 2 — Реляционная модель

	Parent	Child	MAX	MIN
0	Customer	Order	1:N	M-0
1	BookStore	Order	1:N	M-0
2	Order	OrderLineBook	1:N	M-M
3	Book	OrderLineBook	1:N	M-M
4	Book	Author	M:N	0-M

Рисунок 3 — Таблица кардинальностей

4 Заключение

Во время лабораторной работы было выполнено данное по вариантам задание. Результатом нашей деятельности является построение ER-диаграммы, реляционной диаграммы и извлечение таблицы кардинальностей по входным данным.