



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: Куценков Максим и Жовтяк Ярослав

Группа: ИУ9-51Б

Преподаватель: А.Н. Непейвода

2022 г.

СОДЕРЖАНИЕ

1	Техническое задание	3
2	Постановка задачи	4
3	Общий алгоритм работы	5
4	Индивидуальное задание	6
4.1	Входная грамматика	6
4.2	Пользовательский синтаксис	7
4.3	Выходные данные	8
5	Пример работы программы на тестовых данных	9
6	Заключение	14

1 Техническое задание

1. Составить грамматику для описания объекта, указанного в выбранном варианте. Некоторые токены грамматики сделать настраиваемыми параметрами.
2. Составить документацию: как задавать пользовательский синтаксис, а также как записывать требуемый объект.
3. Из отдельного файла *syntax.txt* считать значения параметров. Если файл отсутствует либо в нём не хватает данных, взять значения по умолчанию.
4. Построение объекта по данным, учитывая синтаксис

2 Постановка задачи

- Необходимо предложить грамматику описания сущностей, атрибутов и связей между ними с учётом кардинальностей. Параметризованными токенами грамматиками могут выступать, например:
 - обозначения кардинальностей и типов ключей;
 - 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.

3 Общий алгоритм работы

1. Парсятся файлы, переданные программе в качестве параметров.
2. Создаются внутренние представления информации из файлов.
3. Производятся изменения внутренних представлений, создаются ER и реляционные представления.
4. Внутренние представления переводятся в формат YML, который необходим для создания диаграмм с помощью `eralchemy`.

4 Индивидуальное задание

4.1 Входная грамматика

Входные данные: путь к папке, в которой лежат файлы *test.txt* и, опционально, файл *syntax.txt*. Либо пути к файлам *testname.txt* и *syntaxname.txt* по отдельности. В этих файлах описаны сущности, связи между ними и синтаксис их записи, соответственно.

Входную грамматику можно увидеть ниже:

$$\begin{aligned} [DB] &\rightarrow [Tables] \\ [Tables] &\rightarrow [Table] \mid [Table][TableDelimiter][Tables] \\ [Table] &\rightarrow [Name][LeftParenthesis][TableContents][RightParenthesis] \\ [TableContents] &\rightarrow [ID][Attributes][Connections] \\ [ID] &\rightarrow [IDDenomination][LeftParenthesis][Attributes][RightParenthesis] \\ [Attributes] &\rightarrow [Attribute] \mid [Attribute][AttributeDelimiter][Attributes] \\ [Attribute] &\rightarrow [Name][AttributeTypeDelimiter][AttributeType] \\ [Connections] &\rightarrow [ConnectionDenomination][Name][ConnectionType] \\ [Connection] &\rightarrow [ConnectionDenomination][Name][ConnectionType] \\ [ConnectionType] &\rightarrow [0.1Denomination] \mid [0.NDenomination] \mid [1.1Denomination] \mid [1.NDenomination] \\ [LeftParenthesis] &\rightarrow ' {' \\ [RightParenthesis] &\rightarrow ' } ' \\ [IDDenomination] &\rightarrow ' ID ' \\ [AttributeDelimiter] &\rightarrow ' , ' \\ [ConnectionDelimiter] &\rightarrow ' , ' \\ [Name] &\rightarrow [A - Za - z][NameRight] \\ [NameRight] &\rightarrow [A - Za - z][NameRight] \mid \epsilon \\ [AttributeType] &\rightarrow ' int ' \mid ' float ' \mid ' money ' \mid ' datetime ' \mid ' char ' \mid ' string ' \end{aligned}$$

4.2 Пользовательский синтаксис

- TableDelimiter - разделитель между таблицами
- AttributeTypeDelimiter - разделитель между именем атрибута и его типом
- ConnectionDenomination - ключевое слово для обозначения связи между таблицами
- 0.1Denomination - обозначение кардиальности 0.1
- 0.NDenomination - обозначение кардиальности 0.N
- 1.1Denomination - обозначение кардиальности 1.1
- 1.NDenomination - обозначение кардиальности 1.N

Значения по умолчанию:

- TableDelimiter = ,
- AttributeTypeDelimiter = :
- ConnectionDenomination = CONNECTION
- 0.1Denomination = 0.1
- 0.NDenomination = 0.N
- 1.1Denomination = 1.1
- 1.NDenomination = 1.N

Строки в файле *syntax.txt* разделяются знаком ~. Параметры задаются строками вида *название_параметра = значение* ~. Нельзя использовать строковые значения с знаками = и ~.

4.3 Выходные данные

После запуска программы создается папка *output*, в которой создаются файлы *er_diagram.pdf* и *relational_diagram.pdf*, в которых сгенерированы диаграммы "сущность-связь" и реляционные диаграммы соответственно.

5 Пример работы программы на тестовых данных

Результаты работы при использовании теста из папки *test3*.

test.txt

```
Store {
  ID {
    Location :: String
  }
  Work_Hours :: String ,
  Phone_Number :: String ,
  Capacity :: float ,
  WITH Goods ON,
  WITH Pet ON,
  WITH Order ON
}
NEXT
Goods {
  ID {
    Article_Number :: int ,
    Batch_Number :: int
  }
  Title :: String ,
  Description :: String ,
  Size :: float ,
  Weight :: float ,
  Exp_date :: datetime ,
  Current_price :: money ,
  WITH Store M1,
  WITH Order ON,
}
NEXT
Pet {
  ID {
```

```

        Vet_Certificate_Number :: int
    }
    Genus :: String ,
    Species :: String ,
    Date_of_Birth :: datetime ,
    Price :: money ,
    WITH Store M1,
    WITH Order O1,
    }
NEXT
    Order {
    ID {
    Trade_date :: datetime ,
    Receipt_number :: int ,
    }
    Total_price :: money ,
    Payment_method :: int ,
    WITH Store M1,
    WITH Pet ON,
    WITH Goods ON,
    WITH Employee M1,
    WITH Consumer M1,
    }
NEXT
    Employee {
    ID {
    Employee_id :: int ,
    }
    Name :: String ,
    Surname:: String ,
    Patronymic :: String ,
    Position :: int ,
    WITH Order ON,

```

```

}
NEXT
Consumer {
ID {
E_mail :: String ,
}
Name :: String ,
Surname :: String ,
Phone_Number :: String ,
Bonuses :: int ,
WITH Order ON,
}

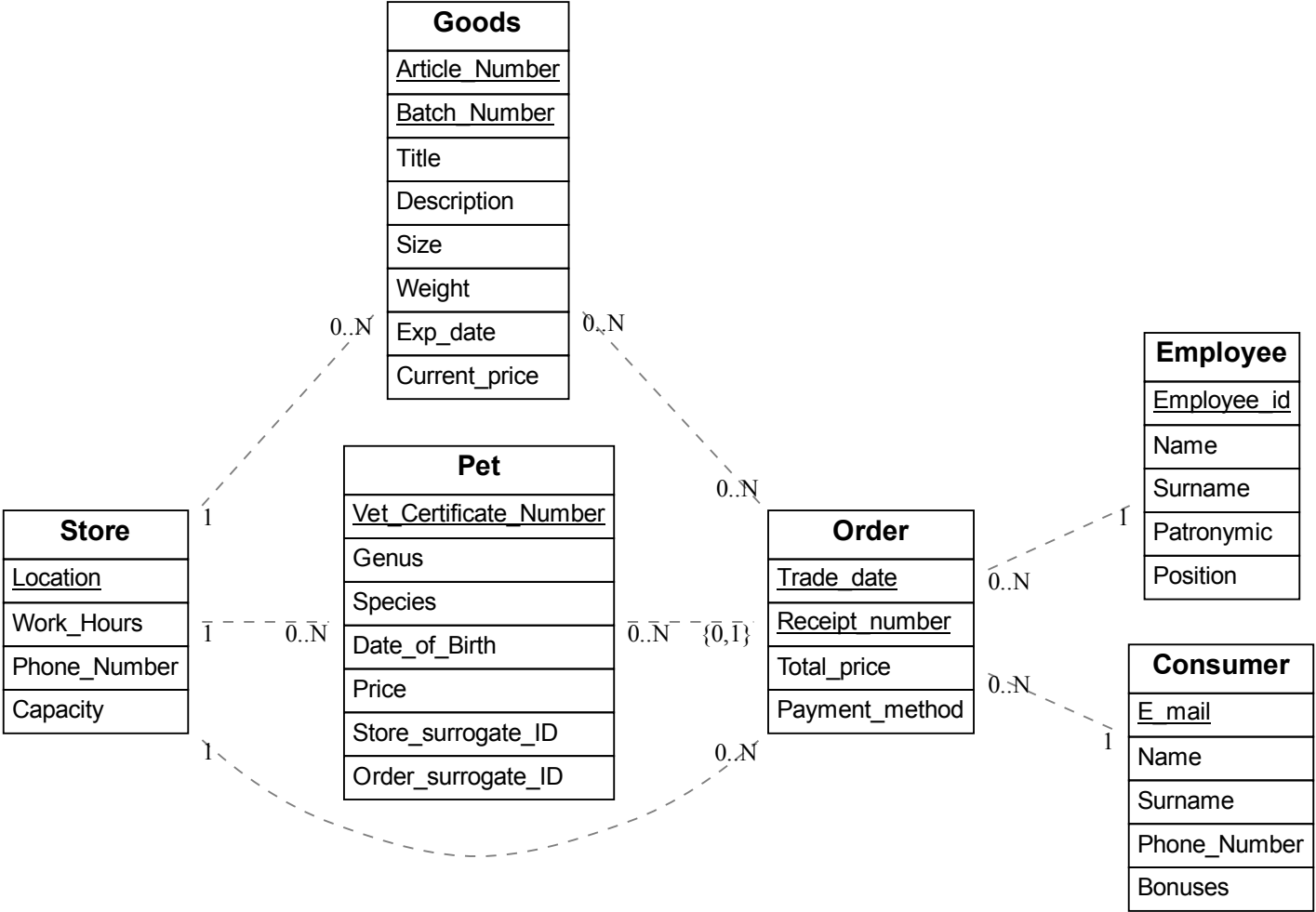
```

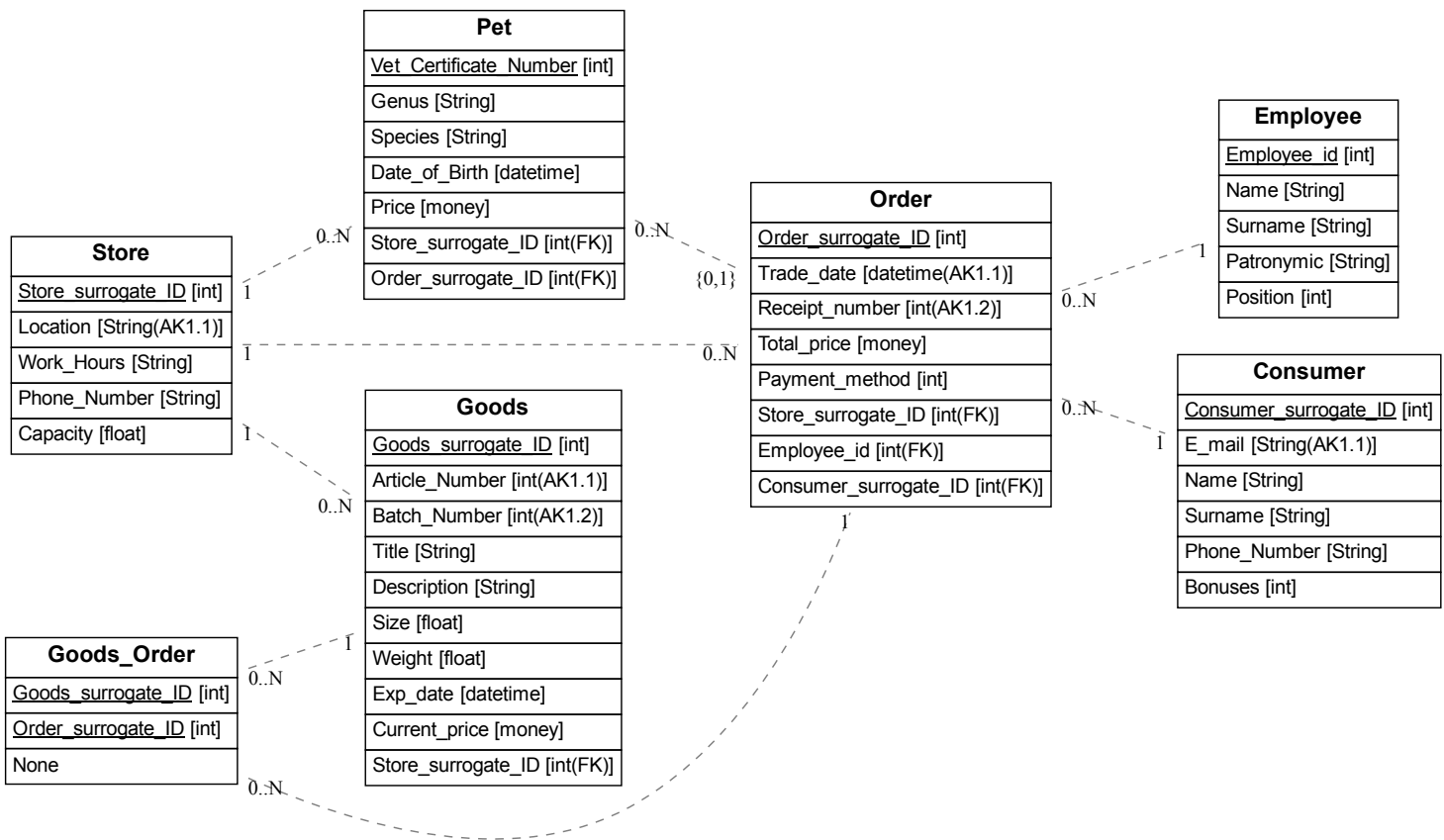
syntax.txt

```

TableDelimiter = NEXT~
AttributeTypeDelimiter = :: ~
ConnectionDenomination = WITH ~
0.1 Denomination = O1 ~
0.N Denomination = ON ~
1.1 Denomination = M1 ~
1.N Denomination = MN

```





6 Заключение

Во время лабораторной работы был выполнен выбранный вариант лабораторной работы. Результатом нашей деятельности является программа, позволяющая создать диаграмму "сущность-связь" и реляционную диаграмму по входным данным, синтаксис которых может определяться пользователем.