Appendix 7-B. Procedure to load articles

```
Open database
Read (aticles200.csv)
For each Article
   Insert (Article to database)
End For
Close database
```

Appendix 7-C. Sentences tokenization and lemmatization procedure

```
Open database
Read (Article)
For each Sentence
        Tokenization (Sentence)
        For each Token
                Lemmatization (Token)
        End For
        Write (Sentence)
End For
Close database
```

Appendix 7-D. Sentences POS tagging procedure

```
Open database
Read (Sentence)
For each Sentence
        For each Token
                Pos_Tagging (Token)
        End for
End For
Close database
```

Appendix 7-E. Sentences NER tagging procedure

```
Open database
Read (Sentence)
For each Sentence
        For each Token
        Ner_Tagging (Token)
    End for
End For
Close database
```

Appendix 7-F. Strategy mention procedure

```
Open database
For each Sentence
    Sentence_Tokens = [Ner_tag_token]
    Elements_Strategy_Mention = [VERB, NOUN, ADJECTIVE, ADVERB,
                                 ADVERB COMPARATIVE, ADJECTIVE COMPARATIVE,
                                 ADJECTIVE SUPERLATIVE, ADVERB SUPERLATIVE]
    If len (val For val in Sentence_Tokens If val in Elements_Strategy_Mention) > 0
        Strategy_Mention = Sentence
        Write (Strategy_Mention)
    End If
End For
Close database
```

Appendix 7-G. Candidate strategies extraction procedure

```
Open database
For each Strategy_Mention
    Sentence_Tokens = [Ner_tag_token]
    If (Sentence_Tokens CONTAINS “VERB” or “NOUN VERB” or “ADJECTIVE
      NOUN VERB” or “NOUN VERB ADVERB”)
        Candidate_Strategy = TokenSelected (Sentence_Tokens)
        Write (Candidate_Strategy)
    End If
Close database
```

Appendix 7-H. Procedure to load strategies of the QEL (sales business)

```
Open database
Read (QEL.csv)
For each QEL
  Insert (Strategy_QEL)
End For
```

```
Close database
```

Appendix 7-I. Procedure to verify the strategies structure

```
Open Database
Read (Candidate_Strategy)
Read (Strategy_QEL)
For each Candidate_Strategy
    Rul_name = []
    If len (Candidate_Strategy == 1) Rul_name.Append (VERB)
    If len (Candidate_Strategy == 2) Rul_name.Append (NOUM VERB)
    If len (Candidate_Strategy == 3) Rul_name.Append (NOUN VERB NOUN / ADJECTIVE NOUN VERB / NOUN VERB ADVERB)
    Write (Strategy_rule)
End For
For each Strategy_QEL
    Rul_name = []
    If len (Strategy_QEL == 1) Rul_name.Append (VERB)
    If len (Strategy_QEL == 2) Rul_name.Append (NOUM VERB)
    If len (Strategy_QEL == 3) Rul_name.Append (NOUN VERB NOUN / ADJECTIVE NOUN VERB / NOUN VERB ADVERB)
    Write (Strategy_rule)
End For
Close database
```

Appendix 7-J. Procedure to weighting of strategies according to the fulfil the business heuristic rules

```
Open database
Read (Strategy_rule)
Read (Heuristic_rule)
TRADING = ['order', 'quotation', 'stock', 'sale', 'price']
DEALING = ['sell', 'buy', 'offer', 'promotion', 'billing', 'cancel']
CRM = ['customer', 'empathy', 'user', 'ecommerce', 'e-commerce', 'omnichannel', 'omni-channel']
For each Strategy_rule
    weight1=0
    If (Strategy_rule == "VERB" or "NOUN") weight1 = 1
    If (Strategy_rule == "NOUN VERB" or "VERB NOUN") weight1 = 2
    If (Strategy_rule == "NOUN VERB NOUN” or “ADEJCTIVE NOUN VERB” or
                                “NOUN VERB ADVERB") weight1=3
    weight2=0
    List = len (val for val in “TRADING” if val in Strategy_rule)
    If len (List) > 0
        Strategy_weight.Append ("TRADING")
        weight2 = weight2 + List*3
    List = len (val for val in “DEALING” if val in Strategy_rule)
    If len (List) > 0
        Strategy_weight.Append ("DEALING")
        weight2 = weight2 + List*3
    List = len (val for val in “CRM” if val in Strategy_rule)
    If len (List) > 0
        Strategy_weight.Append ("CRM")
        weight2 = weight2 + List*3
    weight = (weight1 + weight2) / 12
    Write (Strategy_weight)
End For
Close database
```

Appendix 7-K. Procedure to formalizing the user strategies of sales business

```
Open database
Read (Strategy_weight)
For each Strategy_weight
    If (weight > 0.3)
        Write (User_Strategy)
    End If
End For
```