Translation of Natural Language Queries to SQL Queries towards Building a Natural Language Interface to Database

Submitted by: Tiyasha Kundu

Duration: 28/06/2024 to 28/07/2024

Introduction

1. **Bridge the Gap**

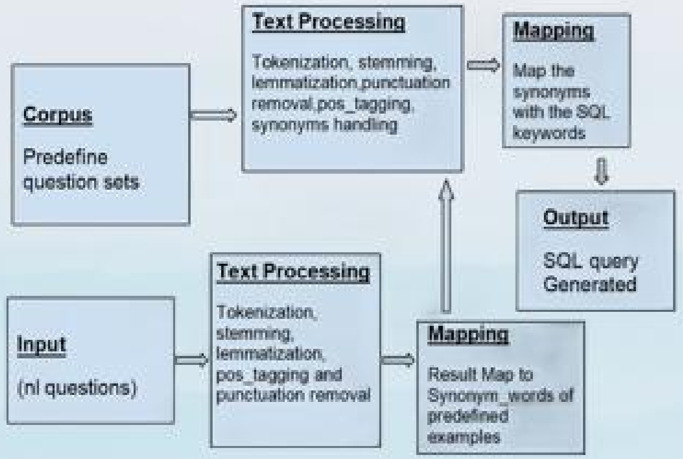
   Aim: Bridge gap between non-technical users and databases.

2. **Focus on NLP**

   Focus: Translate natural language (NL) queries into SQL using NLP techniques like tokenization and POS tagging.

3. **Accessible Queries**

   Goal: Make querying databases more accessible and intuitive.


Corpus
Predefine question sets
Text Processing
Tokenization, stemming, lemmatization,punctuation removal,pos_tagging, synonyms handling
Mapping
Map the synonyms with the SQL keywords
Output
SQL query Generated
Input
(nl questions)
Text Processing
Tokenization, stemming, lemmatization, pos_tagging and punctuation removal
Mapping
Result Map to Synonym_words of predefined examples

Methodology Overview

1

Text Processing

Tokenization, lemmatization, POS tagging.

2

Mapping

Use synonyms to match NL inputs with SQL commands.

3

Corpus Creation

Predefined questions for SQL queries.

Text Processing Methodology in Details

1

Tokenization

Breaks down text into words.

2

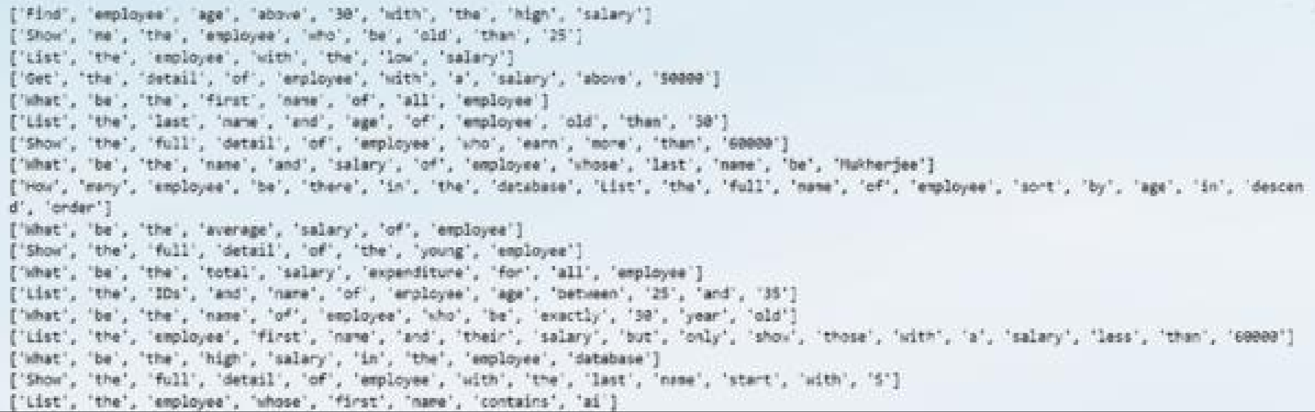
Lemmatization

Reduces words to their base form.

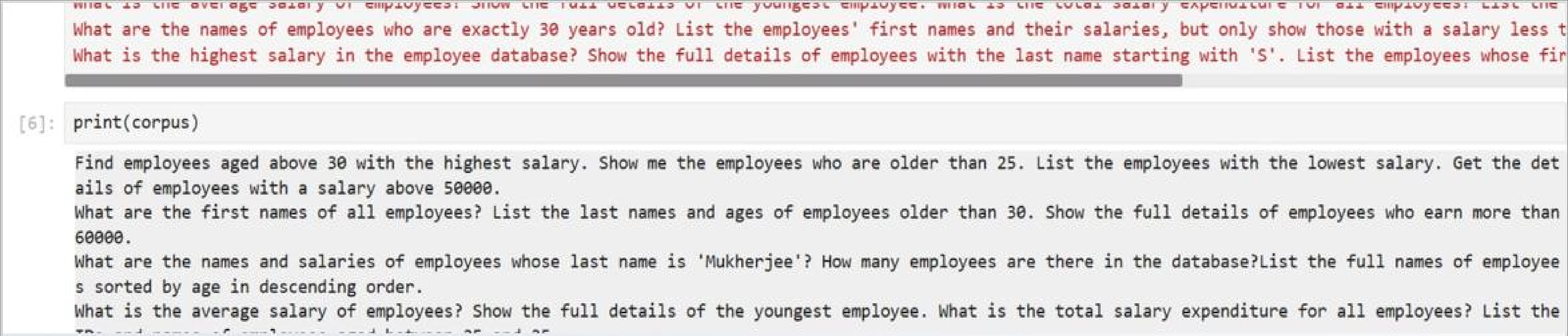
3

Synonym Handling

Maps words to SQL keywords.



```
['Find', 'employee', 'age', 'above', '30', 'with', 'the', 'high', 'salary']
['Show', 'me', 'the', 'employee', 'who', 'be', 'old', 'than', '25']
['List', 'the', 'employee', 'with', 'the', 'low', 'salary']
['Get', 'the', 'detail', 'of', 'employee', 'with', 'a', 'salary', 'above', '50000']
['What', 'be', 'the', 'first', 'name', 'of', 'all', 'employee']
['List', 'the', 'last', 'name', 'and', 'age', 'of', 'employee', 'old', 'than', '30']
['Show', 'the', 'full', 'detail', 'of', 'employee', 'who', 'earn', 'more', 'than', '60000']
['What', 'be', 'the', 'name', 'and', 'salary', 'of', 'employee', 'whose', 'last', 'name', 'be', 'Mukherjee']
['How', 'many', 'employee', 'be', 'there', 'in', 'the', 'database', 'List', 'the', 'full', 'name', 'of', 'employee', 'sort', 'by', 'age', 'in', 'descen
d', 'order']
['What', 'be', 'the', 'average', 'salary', 'of', 'employee']
['Show', 'the', 'full', 'detail', 'of', 'the', 'young', 'employee']
['What', 'be', 'the', 'total', 'salary', 'expenditure', 'for', 'all', 'employee']
['List', 'the', 'IDs', 'and', 'name', 'of', 'employee', 'age', 'between', '25', 'and', '35']
['What', 'be', 'the', 'name', 'of', 'employee', 'who', 'be', 'exactly', '30', 'year', 'old']
['List', 'the', 'employee', 'first', 'name', 'and', 'their', 'salary', 'but', 'only', 'show', 'those', 'with', 'a', 'salary', 'less', 'than', '60000']
['What', 'be', 'the', 'high', 'salary', 'in', 'the', 'employee', 'database']
['Show', 'the', 'full', 'detail', 'of', 'employee', 'with', 'the', 'last', 'name', 'start', 'with', 'S']
['List', 'the', 'employee', 'whose', 'first', 'name', 'contains', 'ai']
```



```
What is the average salary of employees? Show the full details of the youngest employee. What is the total salary expenditure for all employees? List the
What are the names of employees who are exactly 30 years old? List the employees' first names and their salaries, but only show those with a salary less t
What is the highest salary in the employee database? Show the full details of employees with the last name starting with 'S'. List the employees whose fir
```

```
[6]: print(corpus)
```

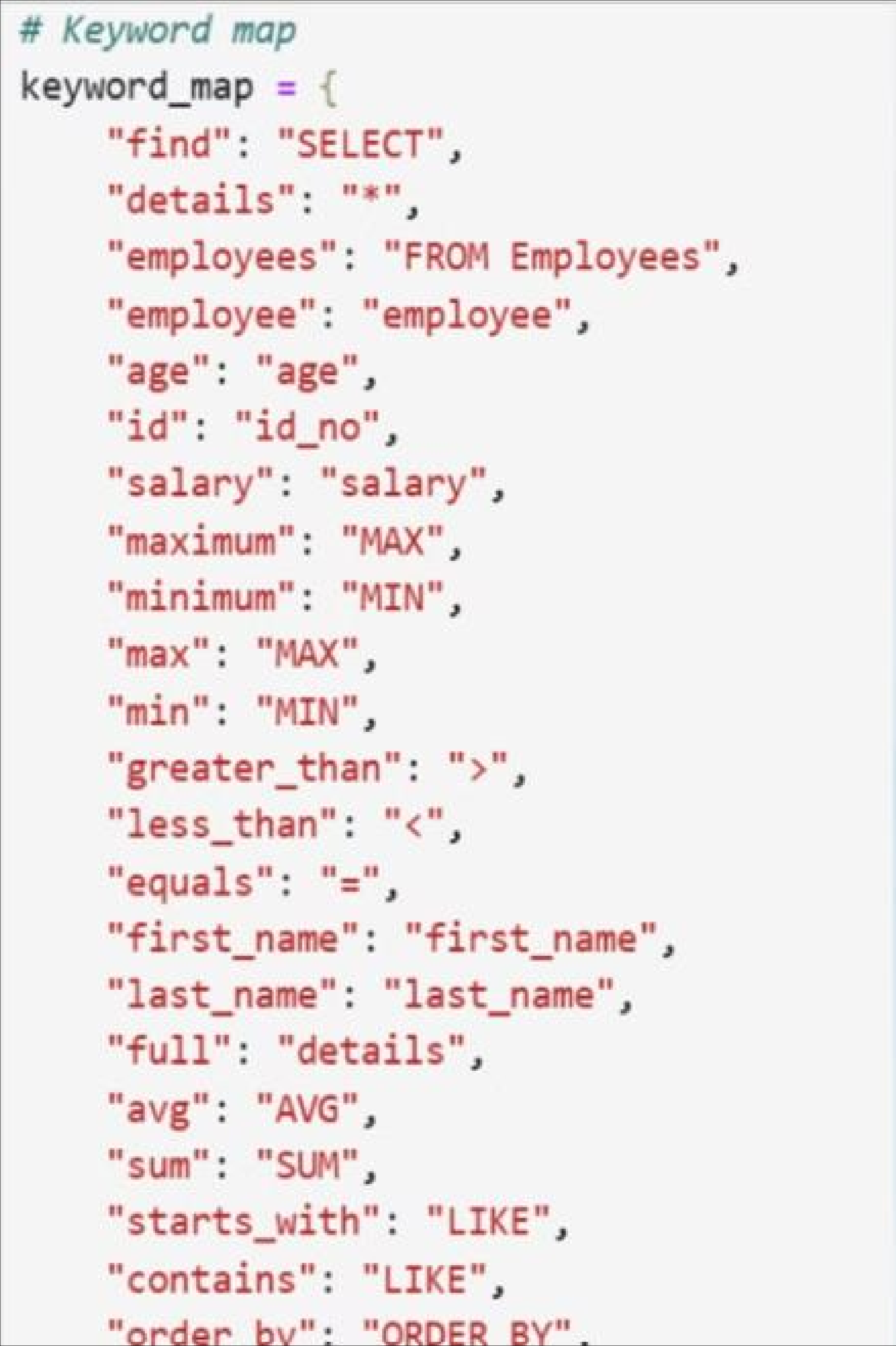
```
Find employees aged above 30 with the highest salary. Show me the employees who are older than 25. List the employees with the lowest salary. Get the det
ails of employees with a salary above 50000.
What are the first names of all employees? List the last names and ages of employees older than 30. Show the full details of employees who earn more than
60000.
What are the names and salaries of employees whose last name is 'Mukherjee'? How many employees are there in the database?List the full names of employee
s sorted by age in descending order.
What is the average salary of employees? Show the full details of the youngest employee. What is the total salary expenditure for all employees? List the
```

Corpus and Queries

Created SQL tables and corresponding questions.

Examples:

Query	SQL
"Find employees aged above 30 with the highest salary."	SELECT * FROM Employees WHERE age > 30 ORDER BY salary DESC;



```python
# Keyword map
keyword_map = {
    "find": "SELECT",
    "details": "*",
    "employees": "FROM Employees",
    "employee": "employee",
    "age": "age",
    "id": "id_no",
    "salary": "salary",
    "maximum": "MAX",
    "minimum": "MIN",
    "max": "MAX",
    "min": "MIN",
    "greater_than": ">",
    "less_than": "<",
    "equals": "=",
    "first_name": "first_name",
    "last_name": "last_name",
    "full": "details",
    "avg": "AVG",
    "sum": "SUM",
    "starts_with": "LIKE",
    "contains": "LIKE",
    "order_by": "ORDER BY",
```

Mapping and Synonym Handling

1. Mapped synonyms to SQL keywords.
2. Automated mapping to reduce manual errors.
3. Used WordNet for a variety of NL sentence inputs.
4. Removed punctuation and stopwords (with issues noted for important words).
5. Used POS tagging to refine SQL mappings.

SQL Query Generation

Function Development

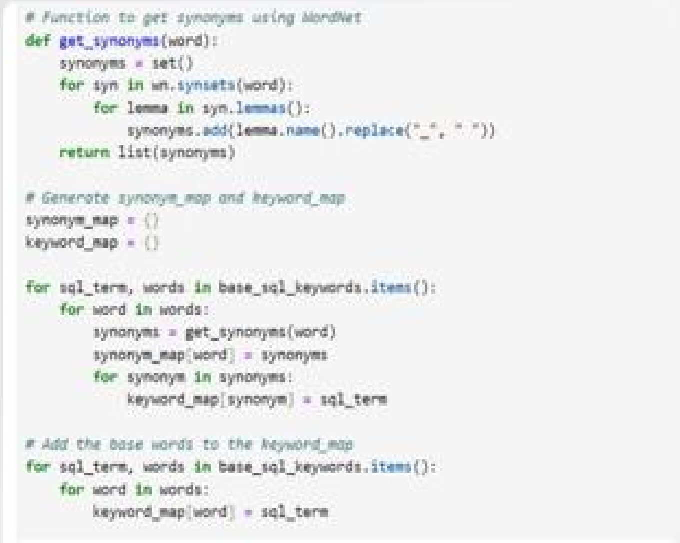
Developed functions for generating SQL queries.

Successful Queries

Out of 18 test queries, 10 were successful.

Output Generation

Example-based and user-based outputs generated.



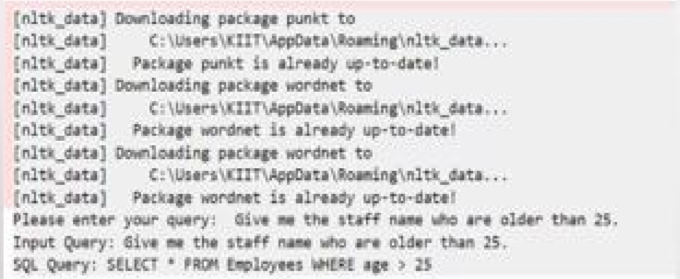
```python
# Function to get synonyms using WordNet
def get_synonyms(word):
    synonyms = set()
    for syn in wn.synsets(word):
        for lemma in syn.lemmas():
            synonyms.add(lemma.name().replace("_", " "))
    return list(synonyms)

# Generate synonym_map and keyword_map
synonym_map = {}
keyword_map = {}

for sql_term, words in base_sql_keywords.items():
    for word in words:
        synonyms = get_synonyms(word)
        synonym_map[word] = synonyms
        for synonym in synonyms:
            keyword_map[synonym] = sql_term

# Add the base words to the keyword_map
for sql_term, words in base_sql_keywords.items():
    for word in words:
        keyword_map[word] = sql_term
```

(figure 3.5)



```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\KIIT\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\KIIT\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\KIIT\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Please enter your query:  Give me the staff name who are older than 25.
Input Query: Give me the staff name who are older than 25.
SQL Query: SELECT * FROM Employees WHERE age > 25
```

(figure 3.6)

Internship Experience, Challenges & Solutions

Delegated Tasks

Designe and implemente the NLID system

Working on query processing, translation, testing, and validation.

Challenges

Building a comprehensive dictionary.

Had to use flexible parsing rules for complex queries.

Performance

Optimized algorithms for better response times.

Tools & Technologies

Python

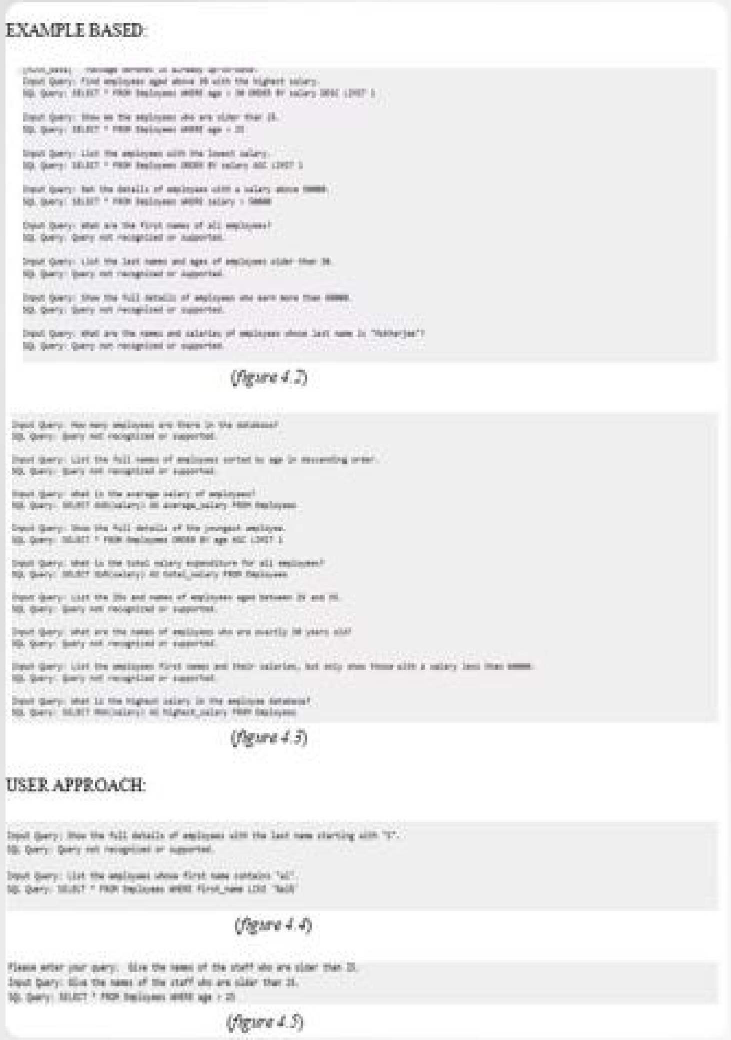
Languages: Python for NLP algorithms.

NLTK

Libraries: NLTK, spaCy for tokenization, lemmatization, and POS tagging.

SQL

Database: SQL for testing the generated queries.



EXAMPLE BASED:

```
[illegible]
Input Query: Find employees aged above 30 with the highest salary.
SQL Query: SELECT * FROM Employees WHERE age > 30 ORDER BY salary DESC LIMIT 1

Input Query: Show me the employees who are older than 25.
SQL Query: SELECT * FROM Employees WHERE age > 25

Input Query: List the employees with the lowest salary.
SQL Query: SELECT * FROM Employees ORDER BY salary ASC LIMIT 1

Input Query: Get the details of employees with a salary above 50000.
SQL Query: SELECT * FROM Employees WHERE salary > 50000

Input Query: What are the first names of all employees?
SQL Query: Query not recognized or supported.

Input Query: List the last names and ages of employees older than 30.
SQL Query: Query not recognized or supported.

Input Query: Show the full details of employees who earn more than 60000.
SQL Query: Query not recognized or supported.

Input Query: What are the names and salaries of employees whose last name is "Mukherjee"?
SQL Query: Query not recognized or supported.
```

(figure 4.2)

```
Input Query: How many employees are there in the database?
SQL Query: Query not recognized or supported.

Input Query: List the full names of employees sorted by age in descending order.
SQL Query: Query not recognized or supported.

Input Query: What is the average salary of employees?
SQL Query: SELECT AVG(salary) AS average_salary FROM Employees

Input Query: Show the full details of the youngest employee.
SQL Query: SELECT * FROM Employees ORDER BY age ASC LIMIT 1

Input Query: What is the total salary expenditure for all employees?
SQL Query: SELECT SUM(salary) AS total_salary FROM Employees

Input Query: List the IDs and names of employees aged between 25 and 35.
SQL Query: Query not recognized or supported.

Input Query: What are the names of employees who are exactly 30 years old?
SQL Query: Query not recognized or supported.

Input Query: List the employees first names and their salaries, but only show those with a salary less than 60000.
SQL Query: Query not recognized or supported.

Input Query: What is the highest salary in the employee database?
SQL Query: SELECT MAX(salary) AS highest_salary FROM Employees
```

(figure 4.3)

USER APPROACH:

```
Input Query: Show the full details of employees with the last name starting with "S".
SQL Query: Query not recognized or supported.

Input Query: List the employees whose first name contains "al".
SQL Query: SELECT * FROM Employees WHERE first_name LIKE '%al%'
```

(figure 4.4)

```
Please enter your query:  Give the names of the staff who are older than 25.
Input Query: Give the names of the staff who are older than 25.
SQL Query: SELECT * FROM Employees WHERE age > 25
```

(figure 4.5)

Conclusion

1 nced Accessibility

NL2SQL system enhances accessibility for database querying.

2 Future Work

Future work: Connect Python to SQL for real-time queries and expand the system for more complex queries.