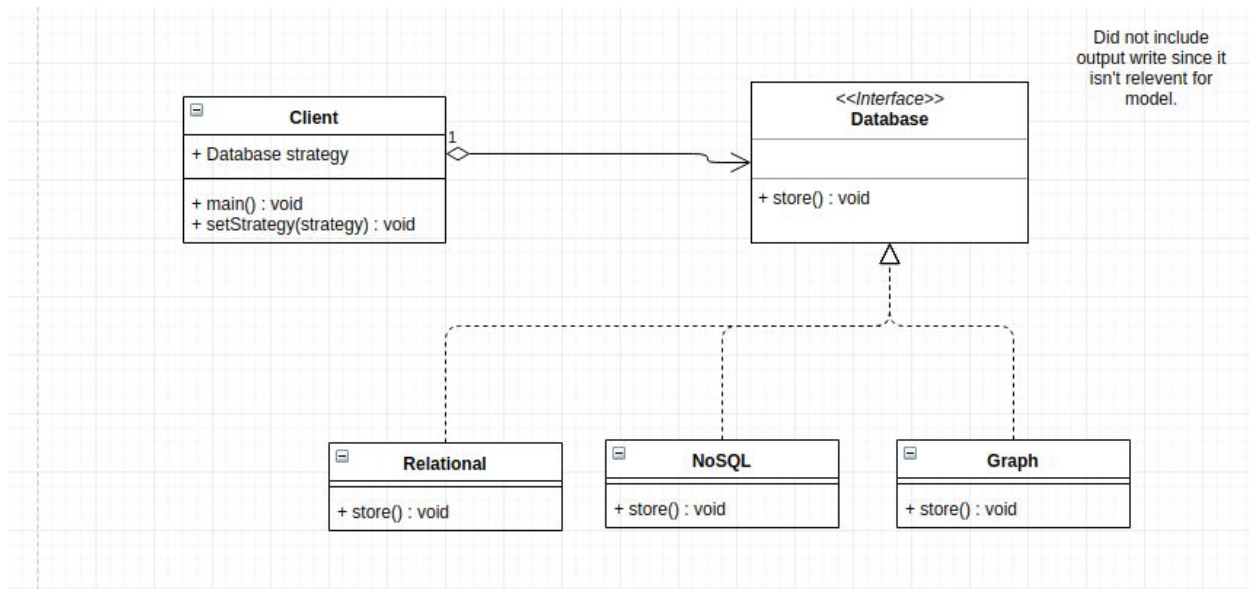


## UML Diagram:



## UML Sequence Diagram:

## Code Output:

```
Console
<terminated> Client [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Sep 19, 2019, 9:41:39 AM)
Please select a Database to start with.
0) Relation
1) Graph
2) NoSQL
2
Document Store has been executed

Please select one of the following options
1) Select a new Database Language
2) End Program
1
0) Relation
1) Graph
2) NoSQL
0
Table Store has been executed

Please select one of the following options
1) Select a new Database Language
2) End Program
1
0) Relation
1) Graph
2) NoSQL
1
Node Store has been executed

Please select one of the following options
1) Select a new Database Language
2) End Program
1
0) Relation
1) Graph
2) NoSQL
2
Document Store has been executed

Please select one of the following options
1) Select a new Database Language
2) End Program
2
Program has Ended!
```

```
out.txt
~/Desktop/Output
Document Store has been executed
Table Store has been executed
Node Store has been executed
Document Store has been executed
```

Code:

```
package homework2V2;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

public class Client {

    // variable for current strategy
    DataBase strategy;

    // This constructor establishes the first strategy
    public Client(DataBase currentMethod) {
        this.strategy = currentMethod;
    }

    public static void main(String[] args) throws IOException {
        // Give the location for the file creation
        String path = "/home/hunter/Desktop/Output/out.txt";

        PrintWriter writer = new PrintWriter(path, "UTF-8");
        Scanner in = new Scanner(System.in);

        // Here are the possible behaviors for the store() method
        DataBase Relation = new Relation();
        DataBase Graph = new Graph();
        DataBase NoSQL = new NoSQL();

        // I put the different behaviors in an array because I thought it would be
        // easier to select that way
        DataBase[] arr = { Relation, Graph, NoSQL };

        // This chunk grabs the user's input and creates the client object
        System.out.println("Please select a Database to start with.");
        System.out.println("0) Relation\n1) Graph\n2) NoSQL");
        int input = in.nextInt();

        Client client = new Client(arr[input]);
        client.strategy.store(writer);
    }
}
```

```

        loop: while (true) {
            System.out.println(
                "Please select one of the following options\n 1) Select a
new Database Language\n 2) End Program");
            input = in.nextInt();
            switch (input) {
                case 1:
                    // Change database type and writes to file
                    System.out.println("0) Relation\n1) Graph\n2) NoSQL");
                    input = in.nextInt();
                    client.setStrategy(arr[input]);
                    client.strategy.store(writer);
                    break;

                case 2:
                    System.out.println("Program has Ended!");
                    break loop;
            }
        }
        writer.close();
        in.close();
    }

    // Sets the strategy
    public void setStrategy(DataBase newMethod) {
        this.strategy = newMethod;
    }
}

package homework2V2;

import java.io.PrintWriter;

public class Graph implements DataBase {

    public void store(PrintWriter writer) {
        // This store strategy will use a BST
        System.out.println("Node Store has been executed\n");
        writer.println("Node Store has been executed");
    }
}

```

```
}
```

```
package homework2V2;
```

```
import java.io.PrintWriter;
```

```
public interface DataBase {
```

```
    public void store(PrintWriter writer);
```

```
}
```

```
package homework2V2;
```

```
import java.io.PrintWriter;
```

```
public class Relation implements DataBase {
```

```
    public void store(PrintWriter writer) {
```

```
        // This storing strategy will use a Hash
```

```
        System.out.println("Table Store has been executed\n");
```

```
        writer.println("Table Store has been executed");
```

```
    }
```

```
}
```

```
package homework2V2;
```

```
import java.io.PrintWriter;
```

```
public class NoSQL implements DataBase {
```

```
    public void store(PrintWriter writer) {
```

```
        // This storing strategy will use an ArrayList
```

```
        System.out.println("Document Store has been executed\n");
```

```
        writer.println("Document Store has been executed");
```

```
    }
```

```
}
```